МИКРОПРОЦЕССОРНЫЕ БИС И ИХ ПРИМЕНЕНИЕ

С. Б. ДОМНИН
Е. А. ИВАНОВ
Л. Л. МУРЕНКО

Средства комплексной отладки микропроцессорных устройств

ЭНЕРГОАТОМИЗДАТ

ОГЛАВЛЕНИЕ

Предисловие 3
Введение 5
Глава 1. **Основы построения отладочных средств** 7
1.1. Микропроцессорные устройства как объект отладки . . . 7
1.2. Определение процесса отладки МПУ 9
1.3. Общие сведения об отладочных средствах 12
1.4. Применение отладочных средств на различных этапах жизненного цикла МПУ 14
1.5. Основные функции и состав отладочных средств 14
1.6. Метод внутрисхемной эмуляции 16
1.7. Архитектура основных узлов внутрисхемных эмуляторов . . 19
1.8. Отладка МПУ с помощью эмулятора 27
1.9. Классификация отладочных средств МПУ 30
1.10. Отладочные программные средства 32
1.11. Аппаратурно-программные средства комплексной отладки МПУ 35
1.12. Архитектура средств комплексной отладки МПУ 39
Глава 2. **Универсальный отладочный комплекс** 55
2.1. Архитектура, технические параметры и основные функциональные возможности 55
2.2. Базовый вычислительный комплекс 59
2.3. Блок эмулятора 8-разрядного микропроцессора КР580ВМ80 64
2.4. Блок эмулятора 8-разрядной однокристальной микроЭВМ КМ1816ВЕ48 77
2.5. Блок эмулятора 16-разрядного микропроцессора КМ1810ВМ86 86
2.6. Блок эмулятора 8-разрядного микропроцессора К1821ВМ85А 96
Глава 3. **Технология программно-аппаратурной отладки МПУ** . . . 104
3.1. Отладка МПУ на базе микропроцессора КР580ВМ80 . . . 104
3.2. Отладка МПУ на базе однокристальной микроЭВМ КМ1816ВЕ48 116
3.3. Отладка МПУ на базе микропроцессора КМ1810ВМ86 . . . 122
Приложение 127
Заключение 144
Список литературы 145

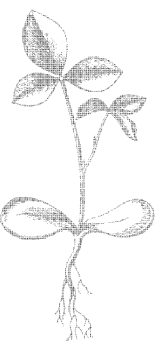
МИКРОПРОЦЕССОРНЫЕ
БИС
И ИХ ПРИМЕНЕНИЕ

С. Б. ДОМНИН
Е. А. ИВАНОВ
Л. Л. МУРЕНКО

Средства комплексной отладки микропроцессорных устройств

Под редакцией доктора
технических наук В. Г. ДОМРАЧЕВА

МОСКВА ЭНЕРГОАТОМИЗДАТ 1988



Scan AAW

ББК 32.97
Д66
УДК 681.322–181.48.001.41

Серия основана в 1988 году

Рецензент И.И. Шагурин

Домнин С.Б. и др.

Д66 Средства комплексной отладки микропроцессорных устройств/ С.Б. Домнин, Е.А. Иванов, Л.Л. Муренко; Под ред. В.Г. Домрачева. – М.: Энергоатомиздат, 1988. – 000 с.: ил. – (Микропроцессорные БИС и их применение).

ISBN 5–283–01490–8

Рассматриваются вопросы проектирования и комплексной отладки аппаратурной и программной компонент микропроцессорных устройств. На конкретных примерах подробно освещены функциональные возможности средств комплексной отладки, особенности их программно-аппаратурной реализации. Приведены рекомендации пользователю в типичных ситуациях.

Для инженеров, занимающихся разработкой и применением микропроцессорной техники.

Д $\frac{2404020000\text{-}290}{051\,(01)\text{-}88}$ КБ-42-13-87 ББК 32.97

Производственное издание

Домнин Сергей Борисович
Иванов Евгений Александрович
Муренко Леонид Леонтьевич

СРЕДСТВА КОМПЛЕКСНОЙ ОТЛАДКИ МИКРОПРОЦЕССОРНЫХ УСТРОЙСТВ

Редактор *А. Д. Иванников.* Редактор издательства *В. И. Петухова*
Художественный редактор *Т. А. Дворецкова, А. А. Белоус*
Технический редактор *Т. Н. Тюрина.* Корректор *С. В. Малышева*

ИБ 2533

Набор выполнен в издательстве. Подписано в печать с оригинала-макета 18.03.88. Т-10520. Формат 60 x 88 1/16. Бумага офсетная № 2. Печать офсетная. Усл. печ.л. 8,82. Усл.кр.-отт. 9,18. Уч.-изд.л. 10,84. Тираж 23 000 экз. Заказ 6392. Цена 55 к.

Энергоатомиздат, 113114, Москва, М-114, Шлюзовая наб., 10.

Отпечатано в ордена Октябрьской Революции и ордена Трудового Красного Знамени МПО "Первая Образцовая типография имени А.А. Жданова" Союзполиграфпрома при Госкомиздате СССР. 113054, Москва, М-54, Валовая, 28.

ISBN 5–283–01490–8

© Энергоатомиздат, 1988

ПРЕДИСЛОВИЕ

В решениях XXVII съезда КПСС провозглашен курс на кардинальное ускорение научно-технического прогресса на основе широкого внедрения новой техники и технологии, комплексной механизации и автоматизации производства и технологических процессов, повышения производительности труда, повышения технического уровня и качества продукции. На современном этапе развития общества решение поставленных задач невозможно без внедрения микропроцессорной техники во всех отраслях народного хозяйства страны. В двенадцатой пятилетке намечено увеличить производство микропроцессорных средств регулирования для систем управления технологическими процессами в 10 раз.

Применение микропроцессорной техники обеспечивает существенный рост производительности труда, улучшение технического уровня и качества продукции, экономию сырья и материалов.

По сравнению с традиционной радиоэлектронной аппаратурой микропроцессорная техника имеет ряд специфических особенностей:

1) высокую сложность и трудоемкость разработки и отладки аппаратурных и в особенности программных компонент микропроцессорных устройств;

2) ограниченное число выводов БИС микропроцессоров и однокристальных микроЭВМ, затрудняющее доступ к их внутренним ресурсам и соответственно поиск и анализ неисправностей в микропроцессорных устройствах;

3) необходимость одновременного контроля за состоянием множества быстропротекающих процессов на внутренних шинах, интерфейсных магистралях и т. п. в процессе отладки, что сильно затрудняет поиск неисправностей в микропроцессорных устройствах.

Для разработки, производства и эксплуатации микропроцессорной техники необходимы специальные инструментальные средства, позволяющие автоматизировать наиболее трудоемкие процессы проектирования и диагностики микропроцессорных устройств – отладочные средства. Среди известных в настоящее время видов отладочных средств важное место занимают средства комплексной отладки, предназначенные для автоматизации проектирования аппаратурных и программных средств и их интеграции в составе микропроцессорных устройств. Для решения этих задач в средствах комплексной отладки широко используются раз-

витые кросс-системы автоматизации программирования, методы программной и аппаратурной эмуляции основных узлов микропроцессорных устройств, методы отыскания и анализа неисправностей в цифровой аппаратуре (трассировка, пошаговый режим, останов по контрольным точкам и т. п.).

Бурное развитие производства персональных ЭВМ в нашей стране и за рубежом позволило сделать этот класс отладочных средств одним из наиболее массовых и распространенных и в то же время существенно расширить его функциональные возможности.

В изданных за последние годы отечественных и переводных книгах по микропроцессорной технике вопросы построения и применения средств комплексной отладки микропроцессорных устройств (МПУ) либо не освещены совсем, либо освещены поверхностно и недостаточно. В особенности все вышесказанное относится к построению, характеристикам и функциональным возможностям отечественных универсальных отладочных комплексов на базе персональных ЭВМ, блоков эмуляторов наиболее распространенных БИС микропроцессоров и однокристальных микроЭВМ, а также алгоритмам функционирования и технологии отладки микропроцессорных устройств с использованием указанных отладочных средств.

Данные вопросы нашли достаточно подробное освещение в предлагаемой книге.

В гл. 1 рассмотрены общие сведения о процессе проектирования и отладки микропроцессорных устройств и построении отладочных средств.

В гл. 2 рассмотрены принципы построения, основные характеристики и функциональные возможности универсального отладочного комплекса на базе персональных ЭВМ типа ДВК, а также блоков эмуляторов 8- и 16-разрядных БИС микропроцессоров и однокристальных микроЭВМ.

В гл. 3 рассмотрена технология программно-аппаратурной отладки МПУ с использованием универсального отладочного комплекса. Представляется, что круг вопросов, освещенный в предлагаемой книге, будет полезен как для разработчиков микропроцессорных устройств, так и для разработчиков средств комплексной отладки.

Авторский коллектив выражает благодарность рецензенту доктору технических наук профессору И.И. Шагурину за внимательный просмотр рукописи и ценные замечания, учет которых несомненно способствовал улучшению книги. Авторы признательны инженерам С.В. Бовину, С.А. Рябову, А.В. Тищенко и А Ю. Уткину за помощь в подготовке материалов, вошедших в гл. 2-ю и 3-ю.

В.Г. Домрачев

ВВЕДЕНИЕ

Микропроцессорными устройствами (МПУ) называют устройства, выполненные на основе схем с большой степенью интеграции (БИС), микропроцессоров (МП) или однокристальных микроЭВМ (ОЭВМ). Микропроцессорные устройства используются практически во всех областях человеческой деятельности, имеют самое разнообразное назначение и выполняют различные функции. При всем многообразии конкретных функций общими функциями МПУ являются управление устройствами ввода и вывода, обработка и преобразование информации.

Основным отличием МПУ от традиционных радиоэлектронных устройств, не содержащих в своем составе МП, является наличие в их структуре двух четко разграниченных категорий средств аппаратурных (АС) и программных (ПС). Микропроцессоры и ОЭВМ придают МПУ свойство программируемости как при разработке, так и при внесении изменений в алгоритм функционирования. Наличие ПС определяет особенности проектирования, производства и эксплуатации, т. е. специфику всего жизненного цикла МПУ.

Программируемость МПУ способствует снижению трудовых затрат, материалоемкости, стоимости, улучшению массогабаритных и энергетических характеристик, повышению надежности и функциональных возможностей устройств. Это обеспечивается благодаря высокой плотности упаковки информации на материальных носителях (различные виды запоминающих устройств), приводящей к большей степени компактности ПС по сравнению с АС аналогичного функционального назначения, а также большой степени универсальности АС МПУ.

В отличие от разработки традиционных радиоэлектронных устройств проектирование МПУ содержит ряд специфических этапов, обусловленных двойственностью их структурного состава (АС и ПС). Основные этапы проектирования МПУ следующие.

Анализ требований к МПУ и структурное проектирование. На этом этапе на основе анализа требований к МПУ, соответствующих специфике конкретных областей применения, разрабатываются общая структура МПУ и алгоритмы его функционирования. Одной из основных задач структурного проектирования является оптимальное разделение МПУ на две функционально завершенные компоненты – АС и ПС, что позволяет далее вести их параллельную разработку.

Проектирование АС МПУ. После принятия компромиссного решения по распределению функций между АС и ПС производится детальное проектирование АС МПУ. В АС любого МПУ можно выделить типовую часть, включающую МП (или ОЭВМ), схемы его тактирования и синхронизации, системный контроллер с шинными формирователями и некоторые другие узлы, и переменную часть, зависящую от конкретных решаемых задач и выполняемых функций. В состав переменной части АС МПУ входят память (ОЗУ, ПЗУ или ППЗУ), интерфейсные схемы сопряжения с внешними устройствами (ВУ) и объектами управления и устройства ввода-вывода (УВВ). Стандартные схемотехнические решения типовой части АС МПУ обычно известны для конкретных типов используемых МП (ОЭВМ), и ими можно воспользоваться при проектировании. Наиболее сложно проектирование переменной части. В целом проектирование АС МПУ, как правило, проще и дешевле, чем проектирование цифровых устройств на ИС малой и средней степени интеграции в связи с относительно небольшим числом БИС в МПУ.

Проектирование ПС МПУ. Этапы проектирования ПС МПУ в основном аналогичны этапам проектирования ПС для ЭВМ, достаточно подробно описанным в литературе. Отметим, что для выполнения этапов программирования и тестирования при проектировании ПС МПУ необходимо использовать специальные инструментальные аппаратурно-программные средства на базе микроЭВМ: резидентные, если МП инструментальной микроЭВМ и МПУ совпадают, и кросс-средства, если МП инструментальной микроЭВМ и МПУ различны. Микропроцессор (ОЭВМ), на базе которого построено МПУ, называют также *целевым* МП (ОЭВМ).

Интеграция АС и ПС МПУ, комплексная отладка и тестирование. Несмотря на то что проектирование отдельно АС и ПС включает в себя этапы их тестирования и отладки, основная трудоемкость приходится на этап комплексной отладки, осуществляемый при непосредственном взаимодействии и взаимосвязи АС и ПС в функционально и конструктивно едином устройстве. Все виды отладки занимают более 50 % полного времени проектирования МПУ. Наличие неизбежных ошибок в программе и необходимость экспериментальной отработки взаимодействия программы с реальным объектом и устройствами ввода-вывода в реальном масштабе времени требуют применения специальных методов и средств комплексной отладки и тестирования МПУ, которые значительно снижают трудоемкость проектирования и соответственно повышают производительность труда разработчика. Более подробно методы и средства комплексной отладки МПУ будут изложены ниже.

После завершения процесса комплексной отладки производится окончательная разработка технической документации на АС и ПС, изготовление опытных образцов МПУ и их испытания.

ГЛАВА 1. ОСНОВЫ ПОСТРОЕНИЯ ОТЛАДОЧНЫХ СРЕДСТВ

1.1. Микропроцессорные устройства как объект отладки

Широкое внедрение МП и ОЭВМ в народном хозяйстве обусловило появление отладочных средств как принципиально нового и весьма специфичного инструмента разработчика МПУ. Очевидно, что особенности характеристик и структуры МПУ целесообразно рассмотреть не только с точки зрения его основного функционального назначения, но и выявить специфику МПУ как объекта отладки. Это позволит более полно и глубоко проанализировать методы и средства отладки. Наиболее важные особенности МПУ следующие [1, 2].

Высокий технологический уровень основной элементной базы МПУ. Применение БИС обеспечивает высокие характеристики при производстве и эксплуатации МПУ. С другой стороны, полный контроль работоспособности таких БИС с помощью стандартной измерительной аппаратуры практически невозможен и требует специальных методов и средств. Особенно затруднены все виды контроля БИС непосредственно в составе МПУ.

Ограниченность физического доступа к внутренним узлам и элементам БИС и ограниченное число контрольных точек. Это в основном связано с ограниченным числом контактных выводов БИС, что в значительной мере затрудняет анализ и выявление в процессе отладки причин ошибок и неисправностей МПУ с помощью стандартных методов и средств.

Тесная взаимосвязь АС и ПС МПУ. Это приводит к трудностям при анализе и выявлении причин ошибок и функциональных неисправностей МПУ, так как большинство выполняемых функций МПУ реализуются аппаратурно программным способом и не поддаются простому разделению на аппаратурную и программную части.

В процессе отладки происходит не только контроль МПУ, но и осуществляется дальнейшее проектирование, отработка МПУ и совершенствование его реальных функциональных характеристик. При этом аппаратурная часть МПУ более стандартна и стабильна, мало подвержена изменению. В процессе отладки изменяется и дорабатывается более гибкая часть МПУ — программные средства.

Существенные архитектурные различия типов МП БИС. Микропроцессоры разных серий различаются по системе команд, структуре, технологии изготовления, средствам ввода-вывода и т. п., что приводит соответственно к аналогичным различиям МПУ и препятствует применению единых для всех МПУ универсальных методов и средств контроля и отлад-

ки. Для каждого типа целевого МП (или семейства МП) необходимы свои отладочные средства или по крайней мере модули специализации.

Магистральная организация МПУ. Приводит к необходимости перед комплексной отладкой МПУ проведения тщательного контроля работоспособности магистрали в режиме одновременного контроля состояния всех шин. Кроме того, при отладке МПУ необходимо учитывать тип и специфику магистральных шин конкретного МПУ.

Работа МПУ в реальном времени. Работа с реальным объектом требует проведения его контроля и отладки также в реальных условиях: на реальных рабочих скоростях, с реальными рабочими программами, с реальным потоком данных.

Способность МПУ к самоконтролю и самодиагностике. Наличие в МПУ МП и ПС позволяет использовать их для сбора и обработки информации о состоянии его узлов, блоков и элементов. Встроенные средства самоконтроля и самодиагностики повышают качество и сокращают время отладки МПУ.

Высокая функциональная сложность аппаратурных средств. Обусловленная целевым назначением МПУ как „интеллектуального устройства" и использованием в его составе БИС и СИС, высокая функциональная сложность требует применения специальных методов и средств проверки функционирования и отладки МПУ.

Кроме указанных особенностей МПУ как объекта отладки необходимо отметить, что непосредственно в процессе отладки используется аппаратурный прототип МПУ, который, совпадая в основном схемотехнически и функционально с реальным МПУ, имеет следующие конструктивные отличия:

вместо БИС МП и БИС ППЗУ в прототипе устанавливаются контактные панельки (розетки) под соответствующие корпуса БИС;

БИС МП и БИС ППЗУ в розетки прототипа не устанавливаются;

вместо реального объекта к МПУ подключается его имитатор.

Указанные конструктивные особенности прототипа МПУ позволяют управлять процессом отладки при интеграции АС и ПС со стороны аппаратуры отладочных средств.

С точки зрения архитектуры целевого МП все МПУ можно условно разделить на две группы: МПУ на базе МП с фиксированной системой команд (МП серий К580, К1816, К1820 и др.) и МПУ на базе микропроцессорных микропрограммируемых комплектов (МП серий К589, К1804 и др.). Архитектурные особенности МПУ первой группы, такие, как система команд, разрядность, спецификация внутренней системной шины и т. п., практически однозначно определяются типом целевого МП. Более того, для каждого типа МП с фиксированной системой команд рекомендуется типовая схема его применения. В архитектуре МПУ второй группы такая однозначность отсутствует. Система команд формируется разработчиком МПУ, им же выбирается необходимая разрядность и определяется спецификация внутренней системной шины.

В силу этих причин разработка МПУ первой группы представляет собой более простую и, главное, массовую задачу по сравнению с МПУ второй группы. Кроме того, большая степень однозначности в постановке задачи проектирования МПУ первой группы делает возможным автоматизировать процесс проектирования в значительно более полной мере. В данной книге вопросы построения средств комплексной отладки будут в основном рассматриваться применительно к МП с фиксированной системой команд.

1.2. Определение процесса отладки МПУ

Отладка МПУ – это процесс обнаружения ошибок, их исправления и обеспечения полной работоспособности МПУ. Микропроцессорное устройство считается отлаженным, когда оно полностью функционирует на реальной аппаратуре в реальных условиях.

Процесс отладки является важнейшим этапом проектирования и создания МПУ и ему придается большое значение потому, что это первый этап проверки и отработки принятых решений в реальных условиях функционирования. Если все предварительные этапы проектирования МПУ можно считать абстрактным проектированием МПУ, при котором АС и ПС представляются в виде схем и текстов программ, то этап отладки следует назвать функциональным проектированием в реальных условиях. Цель этого этапа – обеспечение выполнения всех функций МПУ полностью от начала до конца (рис. 1.1). В результате выполнения каж-

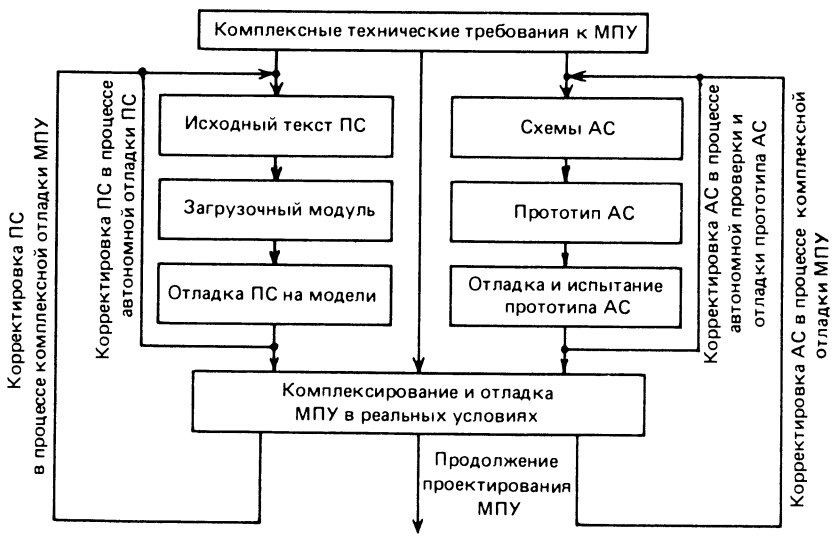


Рис. 1.1. Основные этапы и цикл отладки МПУ

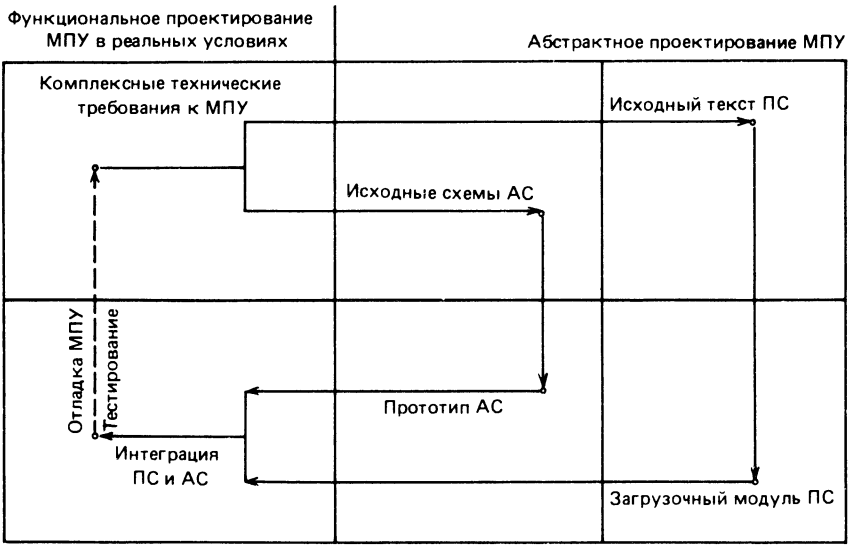


Рис. 1.2. Диаграмма цикла проектирования и отладки МПУ

дой из параллельных ветвей автономного проектирования АС и ПС реализуется лишь часть каждой отдельной функции МПУ, так как согласно сущности МПУ полностью каждая функция может быть выполнена только при комплексном объединении АС и ПС в МПУ.

Процесс отладки МПУ носит циклический характер и придает такой же характер всему процессу проектирования МПУ. На рис. 1.1 показано, что обнаруженные при отладке несоответствия исходным техническим требованиям, некорректное выполнение функций, ошибки и неточности в АС или ПС требуют проведения соответствующих корректировок и по крайней мере частичного повторения всех этапов проектирования. После выполнения корректировок снова необходима отладка, т. е. процесс повторяется. После прохождения каждого цикла отладки МПУ все более удовлетворяет заданным техническим характеристикам. Таким образом, диаграмму цикла проектирования МПУ, представленную на рис. 1.2, с момента начала процесса отладки можно считать диаграммой цикла отладки МПУ.

Цикл отладки включает следующие виды работ:

1) организация взаимодействия АС и ПС в реальных условиях функционирования МПУ;

2) обнаружение несоответствий исходным требованиям, некорректно выполняемых функций, ошибок и неточностей в АС и ПС МПУ и анализ причин и источников их появления;

3) корректировку или доработку прототипа АС и ПС МПУ путем частичного повторения предшествующих этапов проектирования;

4) тестирование МПУ в комплексе АС и ПС;

5) запись отлаженных ПС на носитель в ППЗУ и повторное тестирование МПУ.

Процесс отладки МПУ можно разделить на три вида: автономная отладка АС, автономная отладка ПС и комплексная отладка МПУ [3]. Указанные выше виды работ относятся к комплексной отладке МПУ. Прежде чем приступить к комплексной отладке МПУ, должны быть проведены и отлажены АС и ПС по отдельности.

Автономная отладка АС МПУ. При отладке первоначально обеспечивается работоспособность отдельных устройств МПУ – микропроцессора, ОЗУ, устройств ввода-вывода, магистральных адаптеров и шинных формирователей, устройств сопряжения с объектом, функциональных аппаратных блоков и других устройств. При этом уточняются спецификации и требования к АС МПУ, схемы, чертежи и реальная аппаратура прототипа МПУ. После отладки отдельных устройств МПУ проверяется их взаимодействие. Особо контролируются работоспособность системных магистральных шин данных, адреса и управления (ШД, ША и ШУ) и прием и выдача сигналов на магистральные шины всех связанных магистралью устройств. Проверяются временные диаграммы МП и временные соотношения сигналов МПУ и их комбинаций, форма и уровни импульсов и т. д.

Методы и технические средства, используемые в процессе автономной отладки АС МПУ, в большинстве случаев являются независимыми от типа МП и аналогичны методам и средствам отладки цифровой радиоэлектронной аппаратуры (РЭА) на интегральных схемах с малой степенью интеграции.

Автономная отладка ПС МПУ. При отладке производится тестирование и проверка ПС с помощью программно-логической модели целевого МП на ЭВМ, построенной либо на базе МП того же типа, что и целевой (резидентные средства), либо на базе МП, не совпадающего с целевым МП (кросс-средства). Средства автономной отладки ПС МПУ являются зависимыми от типа целевого МП, т. е. специализированы для каждого типа МП или семейства МП.

Более подробно процесс автономной отладки ПС рассмотрен ниже. Здесь же необходимо отметить, что перед комплексной отладкой МПУ должна быть обязательно проведена автономная отладка ПС, при которой выявляется и исправляется много ошибок и неточностей в программе.

Комплексная отладка МПУ. Отладку выполняют после проведения автономных отладок АС и ПС. При комплексной отладке МПУ происходит интеграция АС и ПС и отлаживаются внешние функции МПУ. Этот вид отладки не является механическим объединением двух предыдущих автономных видов и существенно отличается от них. Комплексная отладка МПУ имеет свои цели, методы и средства, большая часть которых не совпадает с автономными видами отладки. Однако при комплексной отладке приходится, как это указывалось выше, неоднократно проходить цикл проектирования, в процессе которого могут быть использованы ме-

тоды и средства автономных видов отладки. Если методы и средства автономных видов отладки разработаны и существовали до широкого применения МП и МПУ то методы и средства комплексного вида отладки в основном разработаны в связи с широким использованием МП и МПУ в различных сферах человеческой деятельности.

В связи с повышением степени применяемости типовых решений АС в МПУ для каждого типа МП и возрастанием сложности и объема ПС МПУ комплексная отладка имеет большую направленность на отладку ПС в реальной аппаратуре МПУ.

Основное внимание при рассмотрении различных видов отладки целесообразно уделить отладочным средствам и методам комплексной отладки МПУ.

1.3. Общие сведения об отладочных средствах

Для обеспечения и поддержки функционально сложного и трудоемкого процесса отладки МПУ необходимы инструментальные технические средства. Инструментальными средствами в общем смысле этого термина называют вспомогательные технические средства, используемые в процессе разработки изделий. Инструментальные средства, используемые в процессе отладки МПУ, называют *отладочными средствами.*

Отладочные средства МПУ в соответствии с различными видами отладочных процессов подразделяются на три группы (рис. 1.3) : для контроля и отладки АС; для отладки ПС; для комплексной отладки МПУ.

Средства для контроля и отладки АС МПУ. Представляют собой универсальные средства, независимые от типа МП и предназначенные для более широкого применения в дискретной и аналоговой радиоэлектронной аппаратуре. Для отладки АС МПУ на схемном уровне широко используют осциллографы, вольтметры, частотомеры и измерители временных интервалов, генераторы импульсов, простейшие тестеры логического состояния и другие приборы. При выполнении отладки на логическом и функциональном уровнях используют логические анализаторы, сигнатурные анализаторы, комплексы диагностирования и др. Микропроцессор-

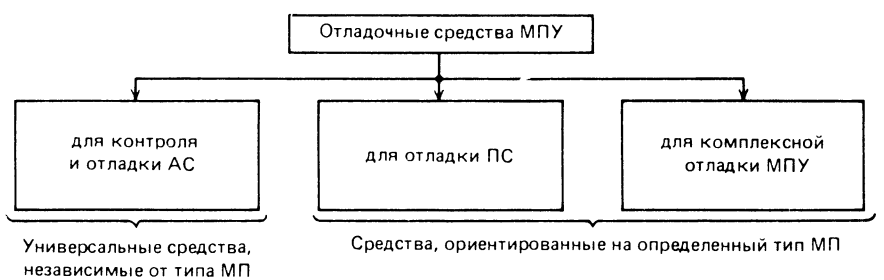


Рис. 1.3. Общая классификация отладочных средств МПУ

ная специфика МПУ не предъявляет дополнительных требований к перечисленным инструментальным средствам, за исключением требований по расширению универсальных функциональных возможностей.

Поскольку основные инструментальные средства контроля и отладки АС являются традиционными для всех РЭА и достаточно хорошо описаны в технической литературе, то здесь они подробно не рассматриваются.

Средства для отладки ПС. Эти средства являются ориентированными на каждый тип МП, предназначенного для использования в МПУ. Так как загрузочные модули ПС для каждого МПУ представлены в машинных кодах целевых МП, системы команд которых и их коды для каждого типа МП различны, то средства для отладки ПС МПУ должны быть ориентированы или специализированы на систему команд каждого типа МП. Средства для отладки ПС представляют собой отладочные программные средства, функционирующие на инструментальной ЭВМ, и подразделяются на резидентные и кросс-средства. Отладочные ПС содержат в своем составе следующие средства: компиляторы с языков высокого уровня, трансляторы с языка АССЕМБЛЕР, редакторы связей, программно-логические модели МП и другие ПС, расширяющие возможности отладки. Компиляторы и трасляторы являются в основном средствами проектирования ПС МПУ, но они включены в состав средств отладки, так как обнаруженные в процессе отладки ошибки в ПС необходимо устранять в исходных текстах программ и производить затем компиляцию или трансляцию ПС. Отладочные ПС могут поставляться как автономно, так и в составе комплексных отладочных средств МПУ.

Средства для комплексной отладки МПУ. Эти средства, так же как и средства отладки ПС, являются ориентированными на конкретный тип МП. предназначенный для использования в составе МПУ. Средства комплексной отладки МПУ имеют следующий состав:

программные средства проектирования и отладки ПС;

аппаратурно-программные средства управления, анализа и контроля функционирования МПУ;

аппаратурно-программные средства записи ПС в ППЗУ;

имитаторы внешней среды.

Необходимо отметить, что в состав средств комплексной отладки МПУ включены средства проектирования и отладки ПС, так как в процессе комплексной отладки с помощью этих средств необходимо проводить устранение ошибок, корректировку и доработку ПС. Кроме того, в процессе комплексной отладки могут быть использованы все средства контроля и отладки АС [13].

Поскольку средства комплексной отладки МПУ имеют наиболее полный функциональный набор инструментальных средств для МПУ, то в дальнейшем этим отладочным средствам будет уделено основное внимание, а сами комплексные средства будут называться отладочными средствами МПУ.

1.4. Применение отладочных средств на различных этапах жизненного цикла МПУ

Применение отладочных средств не ограничивается только этапами разработки и проектирования МПУ. Жизненный цикл МПУ, как и других изделий, включает в себя фазы разработки, производства и эксплуатации. Однако особенность структурного состава МПУ, включающего четко выделенные и в то же время тесно взаимосвязанные АС и ПС, определяет также особенность жизненного цикла МПУ. Наличие ПС в составе МПУ придает ему свойство гибкости при перестройке на новые условия и приводит к необходимости их сопровождения на этапах подготовки производства, производства и эксплуатации МПУ. Причины этого те же, что вызывают необходимость сопровождения вообще любых ПС:

во-первых, в программах, как правило, остаются ошибки, не выявленные в процессе отладки и тестирования;

во-вторых, в ходе подготовки производства, ведения производства и эксплуатации возникает необходимость внесения изменений в ПС и их дальнейшее совершенствование;

в-третьих, практически любые изменения аппаратурной части МПУ в результате изменения технологии или совершенствования характеристик могут вызывать необходимость изменения ПС.

При изменениях ПС или АС на любом этапе жизненного цикла МПУ необходимо использовать отладочные средства. Можно указать следующие случаи применения отладочных средств в процессе производства и эксплуатации МПУ:

доработка МПУ и устранение ошибок проектирования в процессе подготовки производства МПУ;

модернизация МПУ с целью введения новых функций МПУ;

модернизация аппаратуры МПУ, применение новой элементной базы, новых технологий изготовления и т. д.;

перепрограммирование и настройка МПУ на новые внешние условия и объекты;

техническое обслуживание и ремонт МПУ.

1.5. Основные функции и состав отладочных средств

На основе рассмотренных выше видов работ, выполняемых в процессе отладки, можно сформулировать следующие основные функции отладочных средств.

Автоматизация программирования МПУ или разработки ПС. Это очень важная и обязательная функция всех современных отладочных средств, охватывающая этапы проектирования ПС от разработки исходных текстов до получения загрузочного модуля и корректировки и доработки ПС в процессе отладки. Указанная функция отладочных средств обеспечивает автоматизированный переход от высших уровней представления ПС

МПУ к низшим уровням. Автоматизация составления и формирования исходных текстов и дальнейших их преобразований до получения загрузочных модулей является объективно необходимой функцией отладочных средств, так как облегчает наиболее рутинные и формальные этапы труда разработчика МПУ. При этом автоматизация формальных преобразований абстрактных представлений ПС исключает появление дополнительных ошибок, которые неизбежны при ручных способах преобразования, обнаруживает и исправляет некоторые синтаксические ошибки программиста, сокращает время разработки и улучшает условия труда.

Однако может возникнуть вопрос: зачем нужно совмещать традиционные функции проектирования ПС с функциями отладки МПУ? Это необходимо, так как, во-первых, трудно четко разделить процессы проектирования и отладки МПУ и, во-вторых, для исправления ошибок в ПС, их корректировки и доработки в процессе отладки МПУ используются те же методы и средства. Причем следует учесть, что при отладке ошибки в ПС обнаруживаются на функциональном уровне при выполнении программы в виде загрузочного модуля, а исправлять их нужно на высшем уровне – исходных текстов программы и затем повторить весь процесс их формальных преобразований. При разработке ПС и отладке МПУ выполняется ряд следующих этапов: составление и редактирование исходных текстов программ, трансляция, связывание отдельных программных модулей в единый загрузочный модуль, автономное тестирование и отладка ПС и др.

Управление прототипом МПУ при интеграции АС и ПС и комплексной отладке МПУ. Выполнение этой общей функции отладочных средств можно представить в виде следующих конкретных функций: пошагового функционирования МПУ, непрерывное выполнение программы МПУ с заданного адреса, прерывание или остановка функционирования МПУ по заданному условию и др.

Контроль функционирования и регистрации состояния МПУ. Эта функция тесно связана с предыдущей функцией управления МПУ, так как отладочные средства управляют прототипом МПУ с целью его контроля, обнаружения и анализа ошибок функционирования. В процессе контроля функционирования и регистрации состояния МПУ выполняются следующие конкретные функции: регистрация состояния и изменения содержимого памяти и регистров, запоминание и регистрация состояния АС в режиме непрерывного выполнения программы и др.

Запись отлаженных программных средств в БИС ППЗУ. После завершения отладки полностью проверенная и отлаженная программа записывается в программируемые БИС ППЗУ, которые затем устанавливаются в прототип МПУ. Запись в БИС ППЗУ осуществляется на специальном устройстве – программаторе, входящем в состав отладочных средств. Входной информацией для программатора является отлаженный загрузочный модуль на одном из видов машинных носителей. На выходе получается запрограммированная БИС ППЗУ. Программирование осущест-

вляется в автоматическом режиме, чтобы при ручных операциях не внести искажения и ошибки в отлаженную программу.

Перечисленные выше функции определяют следующий обобщенный состав отладочных средств МПУ:

средства разработки и корректировки программ МПУ;

внутрисхемный эмулятор для управления прототипом МПУ;

средства контроля функционирования и регистрации состояний МПУ;

программатор БИС ППЗУ.

Средства разработки и корректировки программ МПУ представляют собой программные средства отладки, которые функционируют совместно с базовым вычислительным комплексом (БВК), включающим устройства отображения информации, дисплей и устройство печати для регистрации информации. Отладочные программные средства ориентированы на систему команд отлаживаемого МПУ и поэтому являются специализированными для каждого типа МП.

Внутрисхемный эмулятор и программатор ППЗУ представляют собой аппаратурно-программные средства. Программатор является универсальным устройством для всех типов стандартных БИС ППЗУ, а внутрисхемный эмулятор в связи с отличиями БИС МП по системе команд, разрядности, структуре, конструкции корпуса и расположению выводов БИС, системам синхронизации, источникам питания, технологии и др. специализирован для каждого типа МП.

Кроме указанных в процессе отладки МПУ используются различные универсальные инструментальные средства, предназначенные для отладки АС МПУ (измерительные приборы, анализаторы, тестеры и др.).

1.6. Метод внутрисхемной эмуляции

Основным методом интеграции АС и ПС и комплексной отладки МПУ является метод внутрисхемной эмуляции. Данный метод заключается в том, что для отладки МПУ создается единая отладочная система, включающая прототип МПУ, объект или имитатор объекта и внутрисхемный эмулятор, которая функционирует как одно целое под управлением эмулятора (рис. 1.4). Внутрисхемный эмулятор, или просто эмулятор, встроенный в общую систему отладки МПУ, дает возможность проводить отладку МПУ или отыскание неисправностей так, как будто эмулятор является частью отлаживаемого МПУ. Итак, *внутрисхемным эмулятором,* или просто *эмулятором,* называют отладочное средство, которое подключается к отлаживаемой или тестируемой системе через микропроцессорный разъем и обеспечивает управление системой путем подмены (эмуляции) отдельных функциональных узлов прототипа МПУ, в первую очередь МП, соответствующими узлами эмулятора.

Эмулятор содержит целевой МП, ОЗУ, устройства ввода-вывода (УВВ), блок управления эмулятором, шину управления и системную шину эмулятора и соединяется с прототипом МПУ с помощью разъема (вилки) с

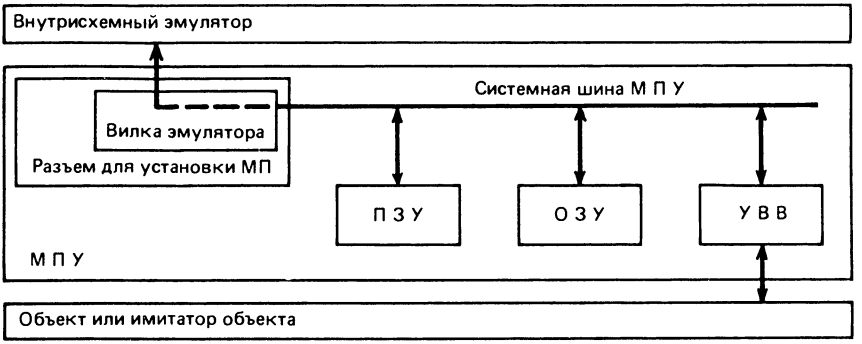


Рис. 1.4. Структурная схема отладочной системы эмулятор – МПУ – объект

двумя рядами параллельно расположенных штыревых контактов, имеющих ту же маркировку и спецификацию, что и целевой МП. Указанный разъем вставляется в гнезда контактной колодки (розетки) прототипа МПУ, предназначенной для установки МП (рис. 1.5). Связанная с эмулятором системная шина МПУ обеспечивает управление всеми аппаратурными средствами прототипа МПУ и не требует никаких дополнительных шин управления. После стыковки разъемов эмулятора и прототипа соединяются их системные шины, и прототип МПУ может функционировать так, как будто в прототип установлен настоящий МП. Однако имеется очень существенное отличие эмулятора от реального МП: при установке в конструктив прототипа реального МП без эмулятора отсутствует возможность контролировать внутреннее состояние и управлять поведением прототипа, за исключением подключаемых к МПУ входных и выходных устройств. С другой стороны, с помощью эмулятора обеспечивается полный контроль состояния МП, гибкое управление прототипом МПУ в различных режимах, необходимых для отладки, а также анализ ошибок в работе МПУ.

Эмулятор в процессе отладки МПУ позволяет эмулировать (заменять) не только целевой МП, но также некоторые функциональные узлы и бло-

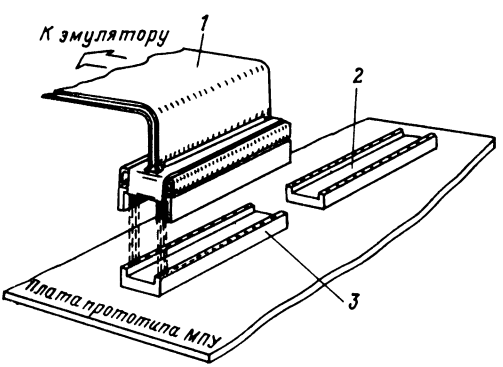


Рис. 1.5. Соединительные колодки для подключения эмулятора к прототипу МПУ:
1 – плоский кабель со штыревым разъемом от эмулятора; *2* – колодка для установки БИС ППЗУ; *3* – колодка для установки БИС МП или подключения эмулятора

Таблица 1.1

Основные стадии отладки МПУ	Аппаратные средства отлаживаемой системы											
	Эмулятор (блок и моделируемые функции)						Прототип (блоки)					
	память программ	устройства ввода-вывода	память данных	блок синхронизации	магистраль системы	целевой МП	разъем целевого МП	магистраль системы	блок синхронизации	память данных	устройства ввода-вывода	память программ
Отладка ПС с реальным МП	+	+	+	+	+	+						
Отладка с реальной магистралью	+	+	+	+		+	+	+				
Отладка с реальным блоком синхронизации	+	+	+			+	+	+	+			
Отладка с реальной памятью данных	+	+				+	+	+	+	+		
Отладка с реальным вводом-выводом	+					+	+	+	+	+	+	
Отладка с запрограммированным ПЗУ						+	+	+	+	+	+	+

Примечание. + – блоки, непосредственно используемые на данной стадии отладки.

ки прототипа МПУ узлами и блоками эмулятора. Например, память эмулятора может быть применена в качестве ОЗУ или ППЗУ прототипа МПУ. Точно так же может быть использован тактовый генератор (блок синхронизации) эмулятора вместо соответствующего блока прототипа. Последовательная подмена частей АС прототипа МПУ, а также широкие функциональные возможности эмулятора по анализу состояния ПС и АС МПУ позволяют производить поэтапную отладку МПУ: от отладки с использованием максимально возможного числа узлов эмулятора постепенно переходят на отладку с использованием полного состава реальных АС прототипа МПУ, за исключением МП (табл. 1.1). При этом для более полного выявления логических ошибок в ПС необходимо по возможности наибольшую часть ПС выполнить и проверить только в составе эмулятора без подключения прототипа МПУ. После выявления ошибок на этом этапе следует переходить к другим этапам отладки МПУ.

1.7. Архитектура основных узлов внутрисхемных эмуляторов

Обобщенная схема внутрисхемного эмулятора (ВСЭ) представлена на рис. 1.6. Внутрисхемный эмулятор состоит из следующих основных узлов: замещаемый микропроцессор или его функциональный аналог; тактовый генератор или схемы синхронизации; память эмулятора; блок управления; внешний интерфейс связи с управляющим устройством или периферийным оборудованием; буфер связи с отлаживаемым МПУ.

Все узлы связаны между собой внутренней магистралью эмулятора, включающей ШД, ША и ШУ.

Микропроцессор. Микропроцессорная БИС или ее функциональный аналог предназначены непосредственно для реализации функций замещаемого МП МПУ по выполнению программ пользователя и являются центральным узлом ВСЭ.

Микропроцессор ВСЭ может быть выполнен одним из трех способов: в качестве микропроцессора ВСЭ используется МП того же типа, что в отлаживаемом МПУ (целевой МП), или один из его вариантов (эмуляционный или диагностический кристалл); микропроцессор ВСЭ выполняется на основе ИС малой и средней степени интеграции; микропроцессор ВСЭ выполняется на основе секционных микропрограммируемых комплектов БИС.

Способ имитации МП ВСЭ определяется архитектурой замещаемого микропроцессора или ОЭВМ, наличием или отсутствием выведенных наружу ША, ШД и ШУ, а также возможностью программного ввода-вывода содержимого внутренних регистров и флагов состояния МП.

Если микропроцессор имеет выведенные наружу ША, ШД и ШУ, то построение внутрисхемного эмулятора выполняется, как правило, на основе целевого МП и не представляет собой особо трудной задачи. Типичным представителем подобного рода МП является БИС типа КР580ВМ80. Выведенные наружу шины адреса, данных и управления могут быть использованы для контроля и управления процессом эмуляции, программного вывода и модификации содержимого внутренних регистров и флагов состояния этого МП. Здесь и далее под процессом эмуляции понимается процесс выполнения МП ВСЭ отлаживаемой программы МПУ.

Нередки случаи, когда особенности архитектуры целевого МП затрудняют построение ВСЭ на его основе. К числу таких архитектурных особенностей можно отнести: реализацию в МП принципов совмещения

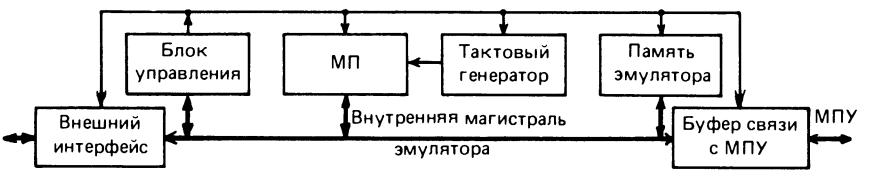


Рис. 1.6. Обобщенная схема внутрисхемного эмулятора

функций и опережающей выборки команд, когда состояние внешних шин МП не отражает характера внутренней обработки информации; отсутствие возможности вывода состояния необходимых внутренних регистров на внешние шины МП; наличие функциональных узлов МП, которые либо не реагируют непосредственно на последовательность выполняемых кодов операций, либо могут менять свое состояние в результате опроса и т. п. [4].

Более сложная ситуация с эмуляцией ОЭВМ. Их отличительной особенностью является наличие на кристалле встроенных каналов ввода-вывода информации, ПЗУ программ и ОЗУ данных. Внутренняя магистраль ОЭВМ или вообще недоступна пользователю, или имеет такой режим работы, при котором внутренняя ША и ШД выводится наружу кристалла, как, например, для семейства серии КМ1816 [5], однако при этом сокращаются ресурсы ОЭВМ, находящиеся в распоряжении пользователя.

Существуют различные способы решения проблемы эмуляции в перечисленных случаях. Один из способов заключается в ведении дополнительных внешних схем, дублирующих некоторые функции целевого МП или ОЭВМ. Способ не связан с большими аппаратными затратами, но не универсален.

Другой способ заключается в построении МП эмулятора на базе цифровых ИС малой и средней степени интеграции. В этом случае разработчик внутрисхемного эмулятора имеет возможность реализовать доступ практически ко всем ресурсам МП или ОЭВМ для анализа их состояния и управления. Недостатки способа – большие аппаратурные затраты – до 700 корпусов ИС и, как следствие, большая потребляемая мощность, высокая стоимость, плохие объемно-массовые показатели. Последнее обстоятельство наиболее существенно, так как ВСЭ представляет собой инструментальное средство разработчика, размещаемое на его рабочем столе совместно с прочей аппаратурой в процессе проведения отладочных или экспериментальных работ.

В последнее время в связи с развитием микроэлектронной технологии, увеличением степени интеграции, повышением быстродействия появилась возможность реализовать МП ВСЭ, обладающий быстродействием целевого МП, на основе секционных микропрограммируемых комплектов БИС. Сохраняя достоинства применения ИС малой и средней степени интеграции, этот способ позволяет снизить аппаратурные затраты на ВСЭ до 300 корпусов. Объемно-массовые показатели, стоимость и потребляемая мощность при этом улучшаются, но признать их полностью удовлетворительными нельзя.

Для эмуляции ОЭВМ может быть использован подход, предусматривающий изготовление на одном кристалле самой ОЭВМ стираемого ППЗУ. Примером его реализации является ОЭВМ 8748 фирмы Intel с встроенным ППЗУ с ультрафиолетовым стиранием, представляющая собой функциональный аналог ОЭВМ 8048. Примером усовершенствованного варианта этого подхода является прибор МК3874 фирмы Mostek

[6] . Основное функциональное назначение таких приборов с точки зрения отладки заключается в их использовании в МПУ вместо эмулируемого МП на завершающем этапе отладки для проверки разработанных программных и аппаратурных средств в реальных условиях. При необходимости коррекция программного обеспечения осуществляется при перезаписи содержимого СППЗУ. Следует отметить, что подобные приборы не заменяют внутрисхемные эмуляторы, а эффективно их дополняют.

Одним из наилучших возможных решений рассматриваемой проблемы является создание специализированных вариантов приборов, предусматривающих доступ к внутренним подсистемам и шинам и позволяющих размещать вне кристалла память программ и данных, обращение к которым обычно производится от внутренних подсистем. В таких приборах специальные контрольные и диагностические точки выводятся на дополнительные контактные площадки; по этой причине подобные компоненты называют диагностическими или эмуляционными кристаллами.

Примером создания специального варианта МП для целей эмуляции является диагностический кристалл 8051 фирмы Intel. Этот диагностический кристалл, заключенный в 64-контактный корпус, не имеет внутренней постоянной памяти. Передача адресов и данных между МП и внешней памятью осуществляется при помощи специализированной шины, выведенной на дополнительные контактные площадки. Диагностический кристалл включает в себя специализированные входные схемы и регистры для управления последовательным портом, таймерами и схемами прерывания в процессе эмуляции [4] .

Контактные площадки эмуляционного кристалла, которые по своему функциональному назначению и временны́м характеристикам идентичны выводам целевого МП, непосредственно подключаются к отлаживаемому МПУ. При этом отсутствует необходимость в каких-либо дополнительных синхронизи́рующих или буферных элементах, что позволяет избежать искажения временны́х соотношений, а также сохранить нагрузочную способность, соответствующую целевому МП.

Такие эмуляционные кристаллы, хотя они обходятся дорого в разработке, представляют собой эффективный инструмент, который позволяет обойти функциональные проблемы, связанные с проведением эмуляции.

Одним из перспективных направлений развития МП является включение в их архитектуру элементов, предназначенных для диагностики и отладки, как это имеет место, например, для МП К1801ВМ2 [7] . Такие диагностические средства должны использоваться преимущественно для отладки и не рекомендуются для эксплуатации у пользователя, так как может возникнуть еще один дополнительный уровень тестирования.

Тактовый генератор. Тактовый генератор (или схемы синхронизации для случая, когда тактовый генератор встроен в МП) обеспечивает работу МП эмулятора на этапе отладки программного обеспечения, когда физический макет МПУ еще не изготовлен.

Тактовый генератор эмулятора настраивается, как правило, на максимальную тактовую частоту целевого МП. С одной стороны, это объясняется тем, что на практике в подавляющем большинстве случаев МП в МПУ работает на максимальной тактовой частоте. С другой стороны, в процессе производства эмулятора обеспечиваются его настройка и проверка в наиболее жестких режимах.

Блок управления с помощью аналогового коммутатора может обеспечивать подключение к МП как внутреннего тактового генератора ВСЭ, так и генератора МПУ. На этапе отладки аппаратурных средств это может быть использовано для диагностики тактового генератора МПУ.

При необходимости тактовый генератор ВСЭ совместно с таймером обеспечивает подсчет времени выполнения отлаживаемых программ или их фрагментов.

Память эмулятора. Основное назначение памяти эмулятора – хранение отлаживаемых программ и данных на этапах отладки как программной, так и аппаратной компонент МПУ.

Обычно отлаживаемые МПУ представляют собой встраиваемые контроллеры, программное обеспечение которых записывается в ПЗУ или ППЗУ. Исправление ошибок в программном обеспечении в этом случае означает замену или перепрограммирование ПЗУ или ППЗУ. Использование оперативной памяти эмулятора для хранения отлаживаемых программ, промежуточных результатов, таблиц данных позволяет быстро и легко вносить изменения и исправления в программное обеспечение и тем самым существенно сократить затраты времени и средств на отладку.

Память эмулятора позволяет провести предварительную проверку отлаживаемой программы в условиях отсутствия макета МПУ. Оперативная память эмулятора может быть использована и для загрузки в нее небольших диагностических программ, выполнение которых облегчает разработчику поиск неисправностей в аппаратуре МПУ.

В памяти эмулятора также размещена служебная программа, на выполнение которой процессор эмулятора переходит после прерывания отлаживаемой программы МПУ при достижении точек останова или выполнения очередного шага программы в пошаговом режиме. Служебная программа является фоновой и предназначена для выполнения команд пользователя: вывода на индикацию информации о состоянии внутренних регистров процессора и содержимого памяти эмулятора и МПУ или изменения их содержимого. Служебная программа может быть размещена как в специальном маскируемом (фоновом) ПЗУ, так и в адресном пространстве памяти программ. Во втором случае на пользователя налагается ограничение по использованию адресного пространства, занимаемого программой. При управлении работой ВСЭ от базового вычислительного комплекса (БВК) это ограничение может быть устранено за счет того, что служебная программа постоянно хранится в памяти БВК, а при достижении МП эмулятора точки останова переписывается для выполнения

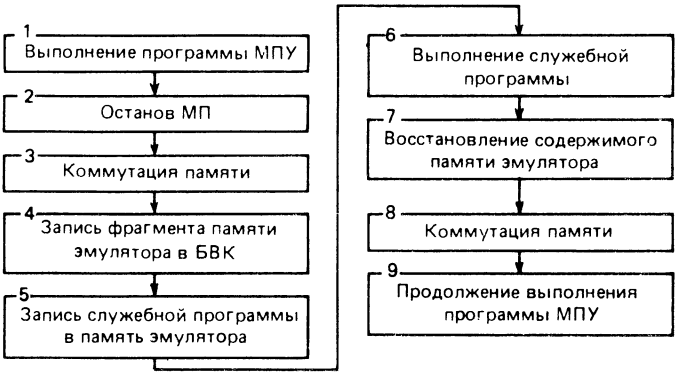


Рис. 1.7. Последовательность действий для выполнения служебной программы, хранимой в памяти БВК

в память эмулятора. При этом соответствующий фрагмент содержимого памяти эмулятора предварительно запоминается в памяти БВК. Последовательность действий для выполнения служебной программы в этом случае представлена на рис. 1.7. Исходное состояние системы – выполнение микропроцессором ВСЭ отлаживаемой программы. При достижении заданного оператором условия МП ВСЭ останавливается по отсутствию сигнала ГОТОВНОСТЬ. Исходно МП ВСЭ может быть подключен как к памяти ВСЭ, так и к памяти МПУ, а служебная программа записывается в память ВСЭ. Следовательно, необходимо скоммутировать память таким образом, чтобы подключить к МП память ВСЭ. При записи служебной программы в память ВСЭ может быть затерта необходимая информация, поэтому предварительно соответствующий фрагмент памяти ВСЭ запоминается в БВК. После записи служебной программы из БВК в память ВСЭ микропроцессор выполняет необходимое число шагов служебной программы. По окончании служебной программы восстанавливают содержимое памяти ВСЭ и ее исходную конфигурацию, на МП подается сигнал ГОТОВНОСТЬ и продолжается выполнение отлаживаемой программы. Указанный алгоритм может быть реализован только для МП, работа которых останавливается по отсутствию сигнала ГОТОВНОСТЬ, что необходимо на время выполнения действий *3*, *4*, *5*, *7*, *8* (рис. 1.7).

Память программ эмулятора может быть дополнена быстродействующей оперативной памятью трассировки или, как ее еще называют, памятью „следа". В памяти трассировки происходит запоминание значений ША, ШД и ШУ, предшествовавших точке останова. Разрядность памяти трассировки определяется архитектурой МП: разрядностью адреса и данных, наличием мультиплексирования адреса и данных или его отсутствия и другими факторами. Объем памяти трассировки невелик и, как правило, не превышает 1–4 К слов, так как пользователю трудно анализировать большие объемы информации.

Память программ и данных эмулятора может быть также использована для организации точек останова (прерывания эмуляции) при обращении МП по заданному адресу. Для этого разрядность ячейки памяти программ и данных эмулятора увеличивается на один разряд, называемый разрядом контрольной точки. Запись в разряд контрольной точки определенного значения, например "1", означает необходимость прерывания эмуляции при обращении МП по заданному адресу. Адресные входы и сигнал выборки памяти контрольных точек объединены с соответствующими входами памяти программ и данных, а выход соединен со входом логики останова эмуляции в блоке управления. По сравнению с методом использования компараторов для прерывания эмуляции данный подход проще в реализации и обеспечивает задание практически неограниченного числа точек останова, однако не позволяет формировать сложных логических условий останова.

Объем отлаживаемых программ МПУ, построенных на МП, редко превышает 16К байт, а для ОЭВМ со встроенной памятью программ составляет 1–4 К байт. По этой причине память программ эмулятора, как правило, организуется в виде банков объемом по 4К байт, пользователю поставляются внутрисхемные эмуляторы с емкостью памяти 4–16 К байт. В более мощных системах пользователь может наращивать эту память в соответствии со своими потребностями.

Проектируя МПУ, пользователь часто использует небольшие объемы памяти, однако распределяет ее в разных областях адресного пространства. Например, собственно программа МПУ размещается в нижней части адресного пространства, а обращение к внешним устройствам (ВУ) в целях упрощения дешифрации осуществляется как к ячейкам памяти, находящимся в старших адресах. Для того чтобы дать возможность пользователю распределить ограниченную память программ эмулятора в адресном пространстве требуемым в таких случаях образом, можно основную часть памяти эмулятора разместить в начальных адресах, а один банк сделать перемещаемым в адресном пространстве. Место этого банка в адресном пространстве задается под управлением монитора в ходе начального диалога при задании конфигурации системы отладки.

При выборе элементной базы для памяти программ эмулятора следует иметь в виду, что к ней предъявляются более жесткие требования по быстродействию, чем к памяти МПУ. Это объясняется тем, что логика блока управления, буфер связи с отлаживаемым МПУ, а также дешифраторы и буфера памяти программ вносят дополнительные задержки в распространение сигналов адреса, данных и выборки.

Блок управления. Блок управления выполняет следующие функции: обеспечивает согласованную во времени работу и коммутацию всех блоков внутрисхемного эмулятора; прием, хранение и дешифрацию командных слов, поступающих от БВК в результате работы управляющей программы; хранение информации о слове состояния и конфигурации аппаратурных средств системы эмулятор–МПУ и их изменение; распо-

знавание событий, вызывающих прерывание выполнения отлаживаемой программы; выполнение служебной (фоновой) программы; управление потоками команд и данных в процессе работы ВСЭ.

Схемы узла распознавания событий во внутрисхемном эмуляторе аналогичны схемам запуска логического анализатора. Различие заключается в том, что в логических анализаторах выходные сигналы этих схем инициируют или прекращают запись данных в память логических последовательностей, а во внутрисхемных эмуляторах они вызывают прерывание выполнения отлаживаемой программы МПУ и переход МП эмулятора на выполнение служебной программы. Переход осуществляется автоматически или под управлением БВК. Чем шире диапазон событий, распознаваемых блоком управления, тем эффективнее работа эмулятора, тем большие возможности и удобства предоставляются пользователю для обнаружения и локализации места неисправности. Условием прерывания эмулятора может быть заданное значение адреса, данных, команды, слова состояния, некоторого внешнего сигнала, их комбинация или последовательность. Подробно схемы распознавания событий рассмотрены при анализе способов запуска логических анализаторов [3].

Эффективным средством отладки, реализуемым в блоке управления большинства внутрисхемных эмуляторов, является картирование памяти (составление карты памяти). Картирование представляет собой процедуру распределения адресного пространства между памятью эмулятора и памятью МПУ, а также назначение отсутствующей в задаваемой конфигурации памяти несмотря на то, что она физически может присутствовать в эмуляторе или МПУ. Составление карты памяти может быть осуществлено с клавиатуры БВК при определении конфигурации системы в начальном диалоге управляющей программы (монитора) или (в более простых устройствах) с помощью переключателей, устанавливаемых на печатной плате внутрисхемного эмулятора.

Пример составления карты памяти представлен на рис. 1.8. Физически в памяти ВСЭ имеются 4 банка ОЗУ, расположенные в адресах с 0000 по 2FFF и с F000 по FFFF. Память МПУ физически реализована в виде бан-

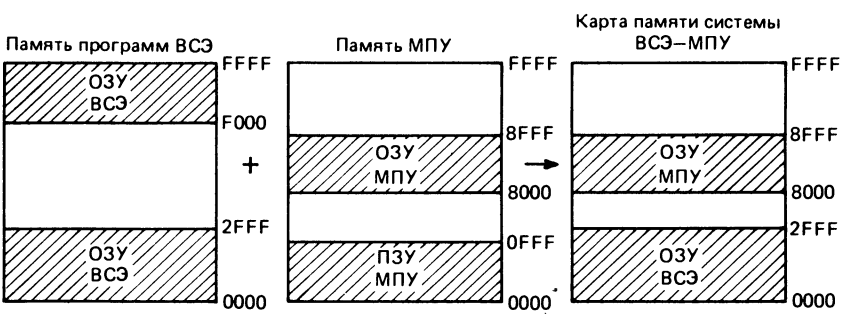


Рис. 1.8. Распределение адресного пространства памяти системы ВСЭ–МПУ

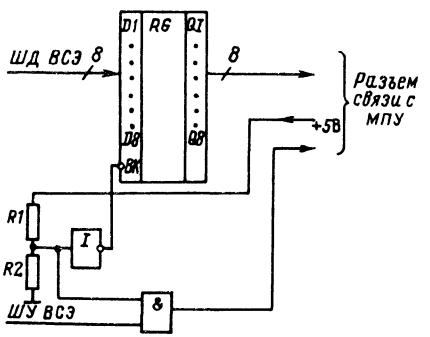
ка ПЗУ в адресах с 0000 по 0FFF и банка ОЗУ в адресах с 8000 по 8FFF. При составлении карты памяти адресное пространство с адреса 0000 по 2FFF объявлено как ОЗУ ВСЭ, а с адреса 8000 по 8FFF как ОЗУ МПУ. По остальным адресам объявлено отсутствие памяти.

Блок управления в процессе выполнения отлаживаемой программы в реальном масштабе времени подключает МП внутрисхемного эмулятора к памяти ВСЭ или МПУ в соответствии с составленной картой памяти. На блок управления может быть возложена функция останова эмуляции при обращении процессора по адресам, объявленным в карте памяти как отсутствующие.

Внешний интерфейс связи. Он обеспечивает передачу управляющих воздействий и обмен информацией между ВСЭ и БВК или периферийным оборудованием. В качестве последнего чаще всего выступает память на магнитных лентах или дисках. Внешний интерфейс служит для организации взаимодействия между внутрисхемным эмулятором и ведущей ЭВМ, их электрического и программного согласования. Схемотехническая реализация внешнего интерфейса проводится в соответствии с установленными рекомендациями и стандартами на выбранный тип интерфейсных шин.

Буфер связи с отлаживаемым МПУ. Буфер предназначен для электрической развязки МПУ и внутренней магистрали ВСЭ, предотвращения сбоев в работе МП при возникновении неисправностей в аппаратурной части отлаживаемого МПУ, а также повышения в случае необходимости нагрузочной способности выходов связи ВСЭ с МПУ. Макет МПУ на начальном этапе отладки может иметь неисправности, которые при подключении внутрисхемного эмулятора могут привести к выходу из строя схем в обоих устройствах.

Чтобы исключить подачу электрических сигналов от эмулятора на МПУ, к которому не подключено напряжение питания, можно использо-



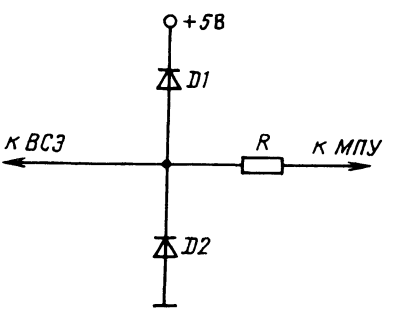


Рис. 1.9. Схема выходных буферов ВСЭ

Рис. 1.10. Схема защиты ВСЭ от подачи запрещенных уровней напряжения со стороны МПУ

вать выходные буфера, включенные по схеме, приведенной на рис. 1.9. Если на МПУ не подано напряжение питания, то низкий уровень напряжения с делителя *R1–R2* запретит прохождение к МПУ сигналов управления через вентиль и переведет буферный регистр данных в третье состояние. Таким образом, на МПУ поступят сигналы низкого уровня или высокоимпедансного состояния. Если же на макет МПУ напряжение питания подано, то через розетку МП МПУ оно поступит на делитель в ВСЭ, который в свою очередь разрешит работу буферных вентилей и регистров.

Защита внутрисхемного эмулятора от подачи запрещенных уровней напряжения со стороны МПУ может быть осуществлена по каждой линии связи в соответствии со схемой, представленной на рис. 1.10.

Общим и определяющим качество ВСЭ требованием, относящимся к построению всех узлов и блоков эмулятора, является „прозрачность" ВСЭ для пользователя. Пользователь по возможности не должен замечать, что МП в МПУ заменен на внутрисхемный эмулятор. Отсутствие полной „прозрачности" ВСЭ определяется, как правило, тем, что часть ресурсов МП отвлекается на организацию функционирования эмулятора. К числу таких ресурсов могут относиться часть адресного пространства, один из уровней прерывания, требование прямого доступа и другие сигналы МП. В результате на пользователя при работе с ВСЭ могут налагаться ограничения по использованию адресного пространства, системы прерывания, средств ввода-вывода, применению некоторых команд, также могут быть внесены искажения во временные соотношения сигналов МП. Очевидно, что чем меньше ограничений налагается на пользователя, чем меньше искажений вносится в работу МПУ, тем выше качество внутрисхемного эмулятора.

1.8. Отладка МПУ с помощью эмулятора

Отладку МПУ с помощью эмулятора можно начинать сразу после того, как произведено подключение МПУ к источникам питания и устранены явные и грубые неисправности (короткие замыкания и обрывы). Однако для уменьшения затрат времени на полную комплексную отладку МПУ целесообразно произвести автономную отладку АС и ПС МПУ. Необходимо особенно тщательно отладить и проверить АС МПУ с помощью имеющихся технических средств (тестеров, логических анализаторов и др.). Уверенность в работоспособности АС сокращает время на обнаружение ошибок в программах.

Основной аксиомой любой отладки является разбиение процесса отладки на ряд по возможности наименьших этапов. Это справедливо также для отладки с помощью эмулятора. Общими для всех случаев являются следующие этапы и методы отладки.

Отладка и контроль АС МПУ. С помощью эмулятора осуществляется последовательное обращение ко всем функциональным узлам и блокам

прототипа МПУ, включая устройства ввода-вывода, и проверяется их работоспособность. При этом целесообразна следующая последовательность проверки: магистральные шины, блок синхронизации (тактовый генератор), память, устройства ввода-вывода и т. д. Например, для проверки связи выходного канала МПУ с внешней средой со стороны эмулятора могут быть переданы данные, адресованные к выходному каналу, который приводится в действие независимо от того, какие конкретно устройства внешней среды подключены к выходному каналу МПУ (индикаторы, реле, сервоусилители и т. п.). Сигналы выходного канала могут быть проконтролированы с помощью простых пробников, тестеров или других измерительных приборов. Аналогично можно адресовать с помощью эмулятора обращение ко входному каналу МПУ и управлять им эмулятором. Для проверки более сложных блоков МПУ может быть составлена простая программа, управляющая работой эмулятора, который в соответствии с этой программой будет периодически адресоваться к входу или выходу МПУ, что создаст циклический источник сигналов для проверки и наблюдения за АС прототипа с помощью осциллографа.

Отладка ПС МПУ и интеграция АС и ПС МПУ. После проверки аппаратуры прототипа можно начинать отладку программы. При этом испытанная и проверенная аппаратура создает реальные условия для успешного выполнения отлаживаемой программы. На этой стадии осуществляется проверка действительных функциональных возможностей или недостатков отлаживаемой программы, а также выявляются и анализируются обусловленные программой ошибки и несоответствия исходным техническим требованиям.

Большинство ПС МПУ должны удовлетворять достаточно жестким требованиям по времени выполнения программы или по быстродействию программ, что соответственно приводит к необходимости отладки и тестирования ПС МПУ в режиме максимального быстродействия МП и МПУ. Однако начальная отладка ПС может осуществляться и при замедленном его выполнении. Для начальной проверки и отладки ПС МПУ существует два основных метода и соответственно режима работы эмулятора: пошаговый и трассировки, т. е. слежения за ходом выполнения программы. При пошаговом методе после выполнения каждой команды программа останавливается и ее выполнение возобновляется по очередной команде оператора. При останове программы эмулятор обеспечивает возможность отображения в удобной форме информации о состоянии АС, задействованных в процессе отладки: отображение содержимого памяти, регистров и т. д. Получаемые результаты сверяются с ожидаемыми, и, если они совпадают, подается команда на выполнение следующего шага программы. Недостатком пошагового метода являются значительные затраты времени. Однако пошаговый метод является эффективным средством предварительной отладки ПС, так как позволяет обнаружить ряд грубых ошибок, которые в режиме непрерывного выполнения могли бы вызвать затирание программы или искажение данных.

При трассировке программы эмулятор обеспечивает выполнение программы в автоматическом режиме, запоминая в памяти или регистрируя на печатающем устройстве содержимое регистров целевого МП после выполнения каждой команды. Некоторые эмуляторы регистрируют также выполняемые команды, представляя их в дизассемблированном виде. При трассировке программы обычно не регистрируется содержимое памяти данных, поэтому в случае необходимости слежения за изменением данных, хранимых в памяти, оператор, отлаживающий МПУ, должен определить содержимое памяти логическим путем. Трассировка программы продолжается автоматически до тех пор, пока не прерывается вручную или по условию останова. Результат трассировки выводится в удобном для восприятия виде на печатающее устройство или экран дисплея. Возможен режим трассировки, при котором регистрируется прохождение МП через заданные контрольные точки программы и запоминается и индицируется состояние МП в этих точках. Выполнение программы от одной контрольной точки до другой происходит в непрерывном режиме.

Трассировка и ее результат в виде зарегистрированных данных являются очень эффективным средством для обнаружения и анализа причин ошибок при выполнении программ. При пользовании этим методом необходимо иметь в виду, что оператор, отлаживающий МПУ, не имеет, кроме останова программы, возможности вмешиваться в процессе трассировки в работу системы и изменять содержимое регистров и памяти данных Следует учитывать также, что при трассировке существует потенциальная опасность затирания отлаживаемой программы или данных при наличии ошибки в программе.

Большое значение для отладки ПС МПУ имеет наличие и использование функции останова (прерывания эмуляции). *Останов* — это прерывание программы, выполняющейся с реальным быстродействием. Место прерывания программы задается в виде конкретного адреса определенной команды (адреса программной памяти) или другого условия. Когда происходит останов, выполнение отлаживаемой программы прекращается и эмулятор переходит на выполнение служебной (фоновой) программы.

Разнообразие условий останова является одной из важнейших технических характеристик ВСЭ. К числу простейших условий останова относятся остановы по кодовому слову. Возможными вариантами могут служить: останов по адресу (команды или данных), останов по заданному коду команды или данных, останов по слову состояния МП, останов по состоянию заданных узлов МПУ и т. д. Наиболее важным и часто употребимым является останов по заданному адресу.

К числу более сложных условий останова относятся остановы по комбинациям кодовых слов. Например, условием останова может быть обращение по некоторому адресу на запись, чтение по заданному адресу определенных данных, обращение по некоторому адресу заданное число

раз. Последнее условие останова может быть весьма эффективно использовано для контроля выхода из циклов, организованных в программе.

Наиболее сложным условием останова является останов по последовательности кодовых слов. Причем условием останова может быть как произвольная последовательность кодов, так и последовательность в заранее определенном порядке. Такой режим останова весьма эффективен для отладки сложных программ. С его помощью можно, например, осуществить контроль за порядком прохождения через определенные ветви программы.

Так же, как и при пошаговом режиме, в случае останова ВСЭ обеспечивает индикацию и модификацию состояния внутренних ресурсов МП, а также памяти ВСЭ и МПУ.

Режим остановов применяется на поздних стадиях отладки, когда риск затирания участков программы при их выполнении между контрольными точками с полным быстродействием является минимальным. Контрольные точки остановов размещаются в местах функционального завершения какого-либо участка программы, где заранее известно или легко определить состояние ее выходных данных (регистры, адреса памяти, устройства ввода-вывода).

Пошаговый метод, трассировка и остановы по контрольным точкам позволяют постепенно охватывать и вовлекать в работу все большую часть ПС МПУ, а средства имитации аппаратуры МПУ в эмуляторе помогают охватывать отладкой последовательно все большую часть АС МПУ.

Для каждого из указанных выше методов отладки эмулятор дает возможность осуществить ряд различных стадий комплексирования аппаратурных средств ВСЭ и МПУ (см. табл. 1.1).

1.9. Классификация отладочных средств МПУ

Подробная классификация отладочных средств представлена на рис. 1.11. В качестве критериев классификации приняты назначение, функционально-конструктивное исполнение, основные эксплуатационные характеристики и широта применяемости в народном хозяйстве. В соответствии с этим отладочные средства подразделяются на три группы средств:

1) отладочные программные средства, которые используются для автоматизации проектирования ПС МПУ и их автономной отладки; эти средства входят в состав комплексных аппаратурно-программных средств отладки, но часто они поставляются и применяются самостоятельно;

2) аппаратурно-программные средства комплексной отладки, предназначенные для интеграции АС и ПС МПУ и их комплексной отладки; эти средства имеют полную функциональную завершенность для поддержки процесса отладки МПУ, включают в свой состав большинство конкретных отладочных средств других групп, выполняют все основные специализированные для отладки МПУ функции и поэтому являются основными отладочными средствами МПУ. Эта группа средств является

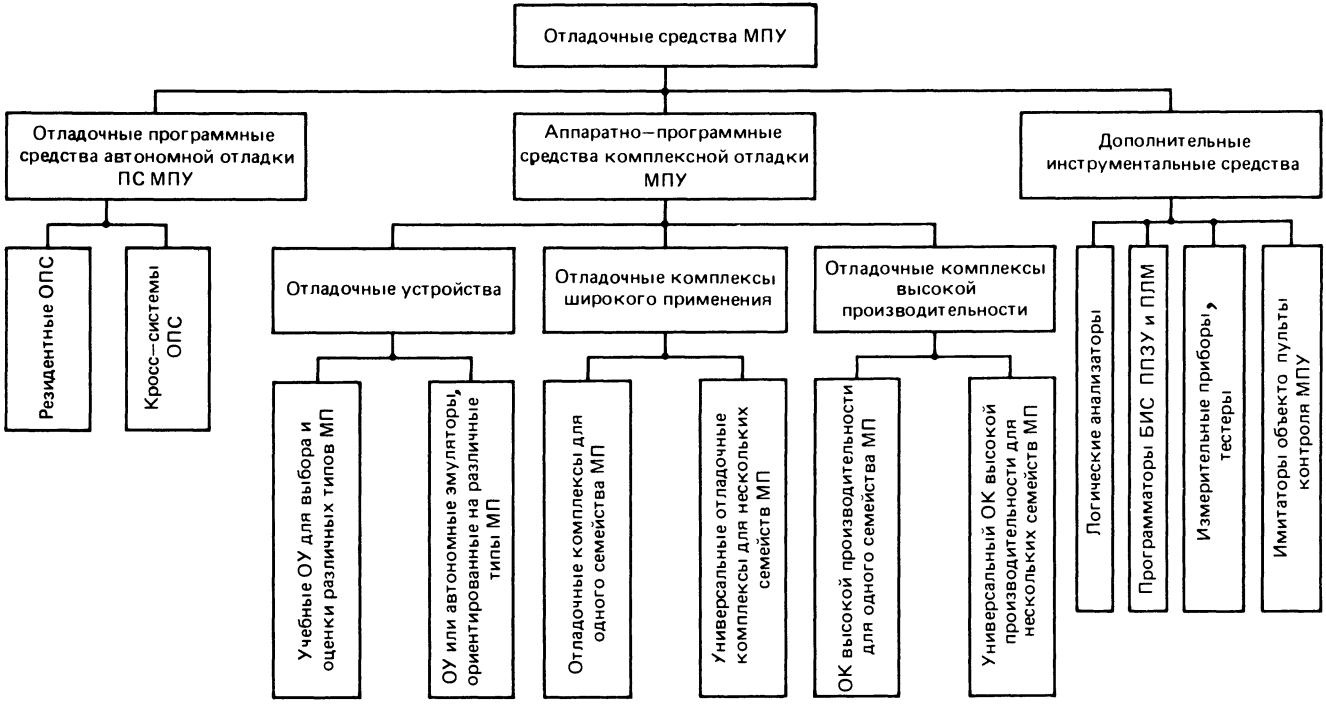


Рис. 1.11. Классификация отладочных средств

самой распространенной и обязательной для отладки всех МПУ. По сложности, стоимости и производительности они могут быть подразделены на три класса средств: отладочные устройства – самые простые, недорогие и массовые, но имеющие самую низкую производительность; отладочные комплексы широкого применения средней сложности, стоимости и производительности, удовлетворяющие требованиям широкого пользователя при создании МПУ средней сложности; отладочные комплексы высокой производительности, обладающие максимальными функциональными возможностями и имеющие соответственно высокую стоимость и узкую область применения для создания сложных МП-систем;

3) дополнительные инструментальные средства, использующиеся в процессе отладки МПУ; эти средства представляют собой универсальные инструментальные средства для контроля, тестирования и отладки дискретных АС, в том числе АС МПУ (измерительные приборы, тестеры, логические анализаторы и др.), а также микропроцессорно независимые вспомогательные средства, использующиеся при отладке МПУ (автономные программаторы БИС ППЗУ и ПЛМ, имитаторы объекта, пульты контроля узлов и блоков МПУ и т. п.).

1.10. Отладочные программные средства

Отладочные программные средства (ОПС) представляют собой комплексы или компоненты программных средств поддержки проектирования, автономной отладки и документирования прикладных пользовательских программ МПУ и функционируют только совместно с аппаратно-программными средствами инструментального базового вычислительного комплекса. Можно сказать, что ОПС функционируют в среде БВК и базовой операционной системы.

В качестве материального носителя ОПС используются носители, предусмотренные в составе БВК (магнитные ленты, диски, ПЗУ). Базовый вычислительный комплекс состоит из микроЭВМ и периферийных устройств: клавиатуры, дисплея, печатающего устройства, внешнего запоминающего устройства в виде накопителей на магнитной ленте, на магнитном диске, полупроводникового модуля ОЗУ большой емкости и др. В качестве примеров БВК можно назвать ЭВМ СМ-4, ЭВМ „Электроника 60", персональные ЭВМ „Электроника ДВК", „Электроника 85" и др. Отладочные ПС выполняются и функционируют в среде определенной операционной системы, которой поддерживается каждый БВК. Операционная система управляет техническими средствами БВК и функционально дополняет их и обеспечивает выполнение следующих функций: управление вводом-выводом, управление хранением информации и обращением к внешней памяти, управление выполнением программ, контроль их выполнения и др. Операционная система БВК совместно с его аппаратурными средствами является ядром ОПС. Схема взаимодействия ОПС с опера-

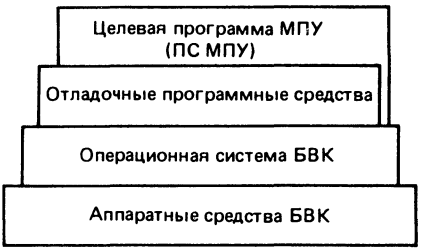


Рис. 1.12. Структурная схема взаимодействия ОПС с программой МПУ и аппаратурно-программными средствами БВК

ционной системой, аппаратными средствами БВК и пользовательской программой МПУ приведена на рис. 1.12. Непосредственно с операционной системой пользовательская программа работает только при вводе (загрузке), составлении, редактировании исходных текстов программы МПУ и выводе преобразованной программы на устройство регистрации, в остальных случаях программа МПУ взаимодействует с операционной системой БВК через ОПС.

Отладочные ПС при взаимодействии с операционной системой выполняют следующие основные функции:

автоматизированное управление составлением, формированием и редактированием исходного текста программы МПУ;

автоматическую трансляцию исходного текста программы МПУ в объектные коды целевого МП или автоматическую компиляцию (при наличии в составе ОПС программы-компилятора с языка высокого уровня) исходного текста программы МПУ, составленного на языке высокого уровня, в объектные коды целевого МП;

редактирование связей программных модулей и подпрограмм, имеющих перемещаемые адреса команд и данных, и компоновка модулей в единый программный загрузочный модуль с абсолютной адресацией, то есть с указанием действительных адресов ПЗУ управляющей памяти МПУ;

программное моделирование или программная имитация системы команд и архитектуры целевого МП для предоставления пользователю возможности отладки загрузочного модуля программы МПУ на программной модели целевого МП (эта функция имеется только у кросс-системы ОПС);

отладку загрузочных модулей целевой программы с реальным процессором БВК, аналогичным по системе команд целевому МП, для резидентной системы ОПС;

накопление и использование библиотеки стандартных программ, модулей и подпрограмм в объектных кодах целевого МП.

Перечисленные функции определяют состав входящих в ОПС программных компонент: программа компилятора (входит не во все ОПС), программно-логическая модель целевого МП (для кросс-систем), программа-отладчик, библиотека стандартных и типовых программ и модулей.

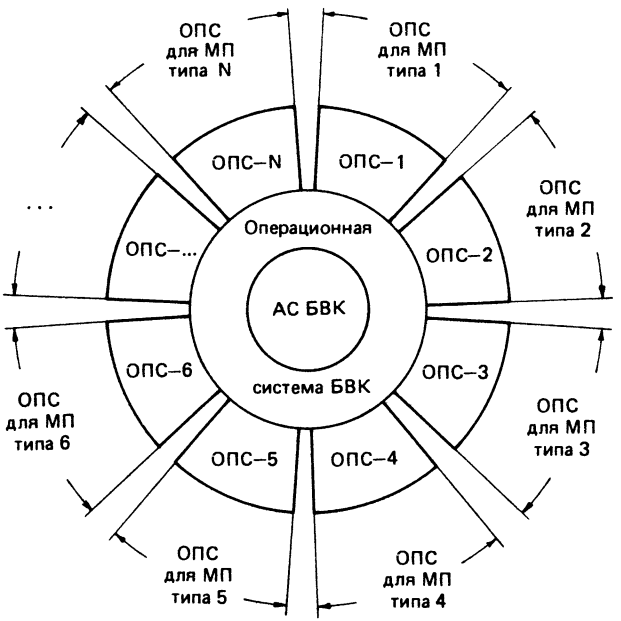


Рис. 1.13. Структура кросс-отладочных программных средств на основе одного БВК для различных типов МП

Резидентные и кросс-системы ОПС по своим функциям и эксплуатационным характеристикам аналогичны и имеют некоторые различия по составу. В состав резидентных ОПС не входит программно-логическая модель целевого МП, вместо которой используется процессор БВК. Кроме того, следует указать, что резидентные ОПС имеют более широкий набор средств программирования для БВК и, следовательно, для целевого МП. Однако число БВК ограничено, и в связи с большим количеством типов выпускаемых целевых МП (4–16-разрядных) не представляется возможным иметь для каждого типа МП резидентные ОПС. Наибольшее распространение имеют кросс-ОПС.

Структурно принципы построения кросс-отладочных средств на основе конкретного БВК для различных типов МП можно представить в виде, указанном на рис. 1.13. Кросс-ОПС являются специализированными средствами как со стороны среды их функционирования – БВК и операционной системы, так и со стороны целевого МП и его системы команд. Если ограничиться одним типом БВК, то для каждого типа МП соответственно существуют различные типы ОПС. На рисунке каждый сектор представляет собой определенный тип ОПС, ориентированный на соответствующий тип МП. Указанное свойство ОПС приводит к увеличению в несколько раз (кратно числу типов применяемых целевых МП) и без

того высокой трудоемкости разработки функционально сложных отладочных программных средств. Однако освоение и применение такой системы в процессе разработки и отладки МПУ значительно повышает производительность, улучшает качество и условия труда пользователя.

Кросс-ОПС обеспечивает возможность написания программы пользователя, автоматического преобразования в объектные коды МП и автономной отладки с помощью программно-логической модели МП. Известно, что любая модель, в том числе программно-логическая, является всего лишь моделью и не может выполнять программу в реальном времени, не может учитывать взаимосвязь между ПС и АС прототипа МПУ. Для устранения этого недостатка необходимо применение аппаратурно-программных средств отладки.

1.11. Аппаратурно-программные средства комплексной отладки МПУ

Наибольшее значение среди всех средств отладки имеют средства комплексной отладки МПУ. Это обусловлено прежде всего важнейшими функциями комплексного этапа отладки: интеграции (комплексирования) АС и ПС МПУ, обнаружения и анализа ошибок, корректировки ПС и АС МПУ, а также максимальными затратами времени на комплексную отладку, достигающими для многих МПУ до 90 % всего времени отладки. Основной целью комплексной отладки является создание полностью функционирующего единого целевого аппаратно-программного МПУ. Как было отмечено выше, процесс комплексной отладки как часть процесса проектирования МПУ является итеративным процессом, каждое последовательное приближение называют *циклом отладки.* Критерии сравнения и оценки конкретных комплексных отладочных средств могут быть сведены к обобщенному критерию эффективности выполнения всего цикла отладки. Очевидно, что разные отладочные средства существенно различаются по способности обеспечения очень трудоемкой задачи неоднократного прохождения отладки. Главное требование заключается в том, чтобы обеспечить минимальные временные и материальные затраты при создании высоконадежных и максимально проверенных АС и ПС МПУ. Эффективность отладочных средств можно определить как предоставление пользователю максимальных функциональных возможностей на всех этапах цикла отладки при минимальной стоимости. С точки зрения пользователя это означает получение максимальных преимуществ от применения отладочных средств за разумную стоимость. Например, нецелесообразно применение высокоэффективных средств, имеющих высокую стоимость, для отладки массовых простых МПУ.

По областям применения, назначению и эффективности средства комплексной отладки МПУ делят на три класса (см. рис. 1.11 и табл. 1.2):

отладочные устройства (в литературе встречаются наименования „оценочные модули", „автономные эмуляторы");

Таблица 1.2

Классы аппаратурно-программных средств комплексной отладки МПУ	Область применения отладочных средств	Степень универсальности отладочных средств	Разрядность целевых МП, бит	Размер программы МПУ, К байт	Производительность отладочных средств и соответствующая стоимость	Конструктивное исполнение отладочных средств
Отладочные устройства	Широкое применение при отладке, ремонте и обслуживании простых МПУ	Специализированные для одного семейства МП	4–8	До 1–2	Малая	Малогабаритный моноблок
Отладочные комплексы широкого применения	Широкое применение при отладке массовых МПУ средней сложности	Специализированные комплексы для одного семейства МП и универсальные комплексы для нескольких семейств МП	8–16	До 64	Средняя	Настольный комплекс аппаратуры
Отладочные комплексы высокой производительности	Ограниченное применение в основном для отладки сложных МП-систем	Специализированные комплексы для одного семейства МП и универсальные комплексы для нескольких семейств МП	8–32	Свыше 64	Высокая	Распределенный комплекс аппаратуры на площади отладочного участка или нескольких участков

отладочные комплексы широкого применения (употребляется также наименование „ микропроцессорная система проектирования”);

отладочные комплексы высокой производительности (употребляются также наименования „микропроцессорная система проектирования”, „интегрированная инструментальная микропроцессорная система проектирования”).

Область применения, показатели назначения, производительность и конструктивное исполнение основных классов средств комплексной отладки приведены в табл. 1.2. Более подробно структура, принципы действия, характеристики средств комплексной отладки будут рассмот-

рены ниже. Необходимо лишь отметить, что наибольшее распространение среди разработчиков МПУ имеют отладочные устройства и отладочные комплексы широкого применения. Причем из этих двух классов отладочных средств с расширением внедрения МП в народное хозяйство благодаря оптимальному сочетанию производительности и стоимости возникает большая потребность в отладочных комплексах широкого применения. Этому способствует, в частности, широкое использование в составе отладочных комплексов средней производительности массовых персональных ЭВМ, обладающих большими функциональными возможностями, развитым и обширным программным обеспечением, полным набором необходимых периферийных устройств, имеющих высокую технологичность массового производства и, как следствие последнего, малую стоимость. Отладочные комплексы высокой производительности используют в своем составе более мощные и дорогие средства вычислительной техники, сложное и дорогостоящее оборудование для отладки МПУ и в связи с этим имеют при максимальных возможностях и характеристиках высокую стоимость, что в значительной степени ограничивает применение этих комплексов только для разработки сложных МП-систем.

Для более четкого определения основных классов средств комплексной отладки необходимо рассмотреть показатель производительности в качестве основной интегральной характеристики отладочных средств. Затраты времени на проектирование существенным образом определяют общую стоимость и технический уровень разработки МПУ. Процесс отладки охватывает все наиболее трудоемкие стадии проектирования МПУ, что обусловливает зависимость затрат времени на проектирование МПУ преимущественно от характеристик отладочных средств. На основании этого можно сформулировать основное требование к инструментальным средствам проектирования МПУ (отладочным средствам) — это обеспечение минимальных затрат времени на проектирование и отладку.

В системе конкретных параметров отладочных средств имеются различные функциональные и временные показатели и параметры, влияющие на общее время отладки МПУ, но отсутствует основной обобщенный показатель, характеризующий затраты времени разработчика МПУ при использовании отладочных средств. Можно предложить такой обобщенный показатель, как производительность отладочных средств, определив его как величину, обратную общему времени, затрачиваемому на проектирование и отладку МПУ. Основные параметры и характеристики, влияющие на производительность средства комплексной отладки различных классов, приведены в табл. 1.3.

Время выполнения каждого этапа проектирования и отладки зависит не только непосредственно от параметров быстродействия технических средств, но в большей степени от их функциональных возможностей. Например, все ОПС, не имея непосредственно временных показателей быстродействия, значительно повышают производительность труда разработчика ПС МПУ и при этом не вносят в процессе автоматического пре-

Таблица 1.3

Наименование показателей назначения, признаков, составных частей средств комплексной отладки	Средства комплексной отладки		
	малой производительности (отладочные устройства)	средней производительности (широкого применения)	высокопроизводительные
Количесто типов целевых МП	Один	Один или несколько	Несколько
Разрядность целевого МП, бит	4–8	8–16	8–32
Размер программы пользователя, К байт	До 2	До 64	Свыше 64
Многопользовательский (многозадачный) режим работы	–	Необязательно	Имеется
Отладка многопроцессорных систем	–	Необязательно	Имеется
Языки программирования (наличие транслятора, компилятора)	Машинные коды или АССЕМБЛЕР	АССЕМБЛЕР, 1–2 языка высокого уровня	АССЕМБЛЕР, несколько языков высокого уровня
Редактор текста	Необязательно	Имеется	Имеется
Редактор связей	Необязательно	Имеется	Имеется
Программа отладчика ПС МПУ	Имеется	Имеется	Имеется
Программа символического отладчика	–	Необязательно	Имеется
Программно-логическая модель	–	Имеется	Имеется
Библиотека стандартных и типовых программных модулей	–	Необязательно	Имеется
Встроенный внутрисхемный эмулятор	Имеется (минимальные функции)	Имеется	Имеется
Функция дизассемблирования	Необязательно	Имеется	Имеется
Выполнение функции загрузки программы пользователя	Вручную	Автоматически	Автоматически
Хранение ПС МПУ при выключении электропитания	Необязательно	Имеется	Имеется
Средства программирования БИС ППЗУ	Необязательно	Имеется	Имеются
Средства анализа и контроля логического состояния АС МПУ	–	Необязательно	Имеются
Средства анализа и контроля временных параметров АС и ПС МПУ	–	Необязательно	Имеются
Стандартные каналы для подключения программируемых измерительных приборов	–	Необязательно	Имеются
Документирование результатов отладки	–	Имеется	Имеется

Характеристики блока управления отладочного устройства или БВК:			
разрядность ЦП	8	16	16–32
дисплей	Однострочный цифровой или алфавитно-цифровой	Алфавитно-цифровой	Алфавитно-цифровой и графический
клавиатура	Цифровая шестнадцатеричная	Стандартная телетайпная	Стандартная телетайпная и графическая
ОЗУ (минимальный объем), К байт	1–2	16, 32, 64	256, 512
объем внешнего ЗУ на магнитных носителях, М байт	–	Не менее 0,4	Не менее 5,0
операционная система	–	Имеется	Имеется
устройства регистрации информации	–	Имеются	Имеются

образования этих ПС дополнительных логических ошибок (при редактировании, трансляции, связывании модулей и др.), что сокращает общее время обнаружения, анализа и корректировки ошибок в программе. Можно указать следующие группы основных характеристик, определяющих производительность комплексных отладочных средств:

функциональные возможности и производительность эмулятора;

функциональные возможности и производительность БВК;

функциональные возможности и быстродействие ОПС;

функциональные возможности отладочного комплекса в целом (многопользовательский режим, отладка микропроцессорных систем и др.);

функциональные возможности вспомогательных устройств отладочного комплекса (анализатор состояний, подсчет времени выполнения программы и др.).

Оценка возможностей комплексных отладочных средств по их производительности позволяет оптимально разрабатывать принципы построения отладочных средств, а также выбирать их для эксплуатации.

1.12. Архитектура средств комплексной отладки МПУ

Как указывалось выше, средства комплексной отладки МПУ подразделяются на следующие основные классы:

отладочные устройства;

отладочные комплексы широкого применения средней производительности;

отладочные комплексы высокой производительности.

Рассмотрим архитектуру указанных классов отладочных средств.

Отладочные устройства. Отладочные устройства (ОУ) имеют наименьшую производительность и используются для комплексной отладки аппаратурных и программных средств простых МПУ, объем управляющих программ которых не превышает 1—4К байт. В этом случае применение внешней ЭВМ с периферийным оборудованием не является обязательным. Отсутствие ЭВМ, с одной стороны, ограничивает функциональные возможности ОУ, но, с другой стороны, резко уменьшает их стоимость, габариты, вес и потребление энергии, значительно повышает надежность работы.

Поскольку ОУ являются автономными устройствами, они должны иметь в своем составе средства для обеспечения диалога с пользователем, в качестве которых чаще всего используются простейшие дисплей и клавиатура. Некоторые типы ОУ могут иметь в своем составе интерфейсные устройства для подключения внешних устройств ввода-вывода (например, кассетный магнитофон, печатающее устройство, программатор для программирования БИС ППЗУ), а также внешней ЭВМ.

Кроме использования на различных этапах разработки и отладки МПУ наиболее простые и дешевые типы ОУ могут применяться также для целей обучения приемам и методам отладки, а также для практического изучения характеристик и возможностей МП при первом знакомстве с ними. Такие ОУ получили название *обучающих отладочных устройств.*

Конструктивно ОУ выполняются в виде малогабаритных настольных моноблоков, некоторые типы ОУ выполняются в виде одноплатных устройств.

По типу внутренней структуры ОУ можно разделить на два класса: с двухпроцессорной и с однопроцессорной структурами.

Особенностью двухпроцессорных ОУ является наличие контроллера на базе некоторого МП, служащего для управления процессом диалога с пользователем, режимами эмуляции и обменом информацией по каналам ввода-вывода, а также второго — целевого МП или однокристальной микроЭВМ, выполняющего программы пользователя и входящего в состав эмулятора.

В однопроцессорных ОУ функции контроллера и целевого МП объединены в одном элементе. Ясно, что двухпроцессорная структура ОУ требует большего состава оборудования, чем однопроцессорная.

Преимуществами двухпроцессорных ОУ являются:

бóльшие функциональные возможности благодаря наличию отдельного контроллера, позволяющего более оптимально реализовать функции управления и обмена информацией. При эмуляции малоразрядных МП и ОЭВМ (4—8 бит) их функциональных возможностей часто оказывается недостаточно для организации необходимого диалога с пользователем и управления;

снятие всех ограничений, налагаемых на технические характеристики эмулятора МП и ОЭВМ, а также на использование всего адресного пространства памяти;

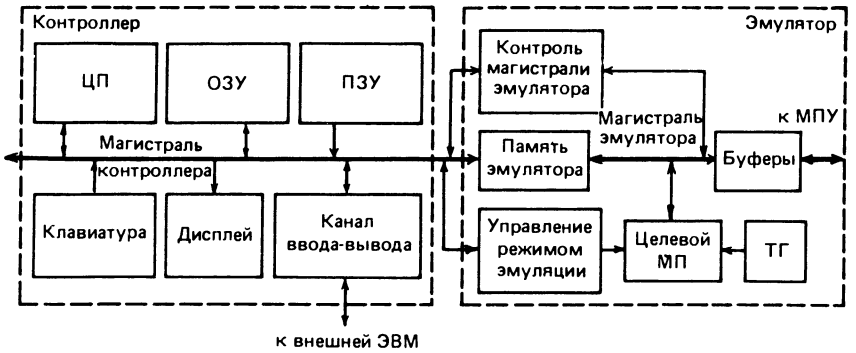


Рис. 1.14. Структурная схема двухпроцессорных отладочных устройств

возможность выделить универсальную часть оборудования, не зависящую от типа целевого МП (например, контроллер), что облегчает проектирование ОУ для нескольких типов МП.

Обобщенная схема двухпроцессорного ОУ приведена на рис. 1.14.

В состав двухпроцессорного ОУ всегда входят два основных функциональных узла: контроллер и эмулятор.

Контроллер выполняет следующие основные функции: осуществление диалога с пользователем с помощью клавиатуры и дисплея; выдача команд управления режимами работы эмулятора; обмен информацией с памятью пользователя; обмен информацией с внешней ЭВМ или внешними устройствами.

В состав контроллера входят: центральный процессор (ЦП); оперативное и постоянное запоминающие устройства (ОЗУ и ПЗУ); клавиатура; дисплей; канал ввода-вывода.

Контроллеры современных двухпроцессорных ОУ конструктивно выполняются чаще всего в виде одноплатной микроЭВМ, содержащей встроенные устройства ввода-вывода (цифровой монодисплей и цифровая клавиатура).

Связь между всеми функциональными узлами контроллера осуществляется с помощью внутренней магистрали контроллера, состав, разрядность и назначение сигналов которой полностью определяются типом и характеристиками ЦП. В качестве ЦП в большинстве случаев применяются 8- и 16-разрядные БИС МП средней и высокой производительности (КР580ВМ80, К1810ВМ86 и др.) Постоянное ЗУ обеспечивает хранение упрвляющей программы-монитора, выполняемой ЦП.

Оперативное ЗУ служит для временного хранения информации, поступающей из внешней ЭВМ и с клавиатуры, а также для хранения информации, выдаваемой в эмулятор и на дисплей. Кроме того, часть ОЗУ может использоваться в качестве рабочей области памяти при выполнении ЦП управляющих программ.

Канал ввода-вывода связывает контроллер с внешней ЭВМ или внешними устройствами, если таковые имеются в распоряжении пользователя. Внешняя ЭВМ позволяет значительно расширить функциональные возможности контроллера благодаря автоматизации процесса разработки и отладки программ МПУ, долговременному хранению подготовленных программ на носителях (магнитная лента, диск и т. п.) и быстрому вводу их в память эмулятора. Внешняя ЭВМ является особенно необходимой в тех случаях, когда отлаживаемая программа является достаточно большой (более 1К байт), а информация в ОЗУ эмулятора разрушается при выключении питания.

Эмулятор выполняет следующие основные функции: эмуляция МП (ОЭВМ), входящего в состав МПУ, благодаря наличию в своем составе целевого МП (ОЭВМ), который является его точной физической моделью; эмуляция памяти МПУ благодаря наличию в составе эмулятора физической модели памяти пользователя; выполнение необходимого набора отладочных функций (пошаговый режим, останов по контрольным точкам, картирование памяти и т. п.); обмен информацией между контроллером и памятью эмулятора; обмен информацией между памятью эмулятора и памятью МПУ.

В рассматриваемой структуре ОУ в состав эмулятора входят: целевой МП (ОЭВМ) с тактовым генератором (ТГ); узел контроля магистрали эмулятора; память эмулятора; узел управления режимами эмуляции; буферы-формирователи.

Связь между всеми функциональными узлами эмулятора осуществляется с помощью магистрали эмулятора, характеристики которой полностью определяются типом целевого МП (ОЭВМ).

Память эмулятора является двухпортовой, т. е. доступ к ней возможен как со стороны контроллера, так и со стороны целевого МП (ОЭВМ). Доступ со стороны контроллера необходим для загрузки памяти эмулятора отлаживаемой программой МПУ, а также для контроля ее содержимого с использованием дисплея. Доступ со стороны целевого МП необходим для загрузки памяти МПУ, а также контроля ее содержимого с помощью дисплея контроллера.

Непосредственное управление доступом к памяти эмулятора, а также режимами эмуляции осуществляется с помощью узла управления режимом эмуляции, получающим по магистрали контроллера управляющие команды.

Узел контроля магистрали эмулятора служит для индикации состояния магистрали эмулятора (а следовательно, и состояния целевого МП), что необходимо для целей отладки и отыскания неисправностей. Управление им также осуществляется от узла управления режимом эмуляции (на структурной схеме с целью упрощения линии управления не показаны).

Наконец, буферы-формирователи увеличивают мощность и служат для электрического согласования магистрали эмулятора с магистралью

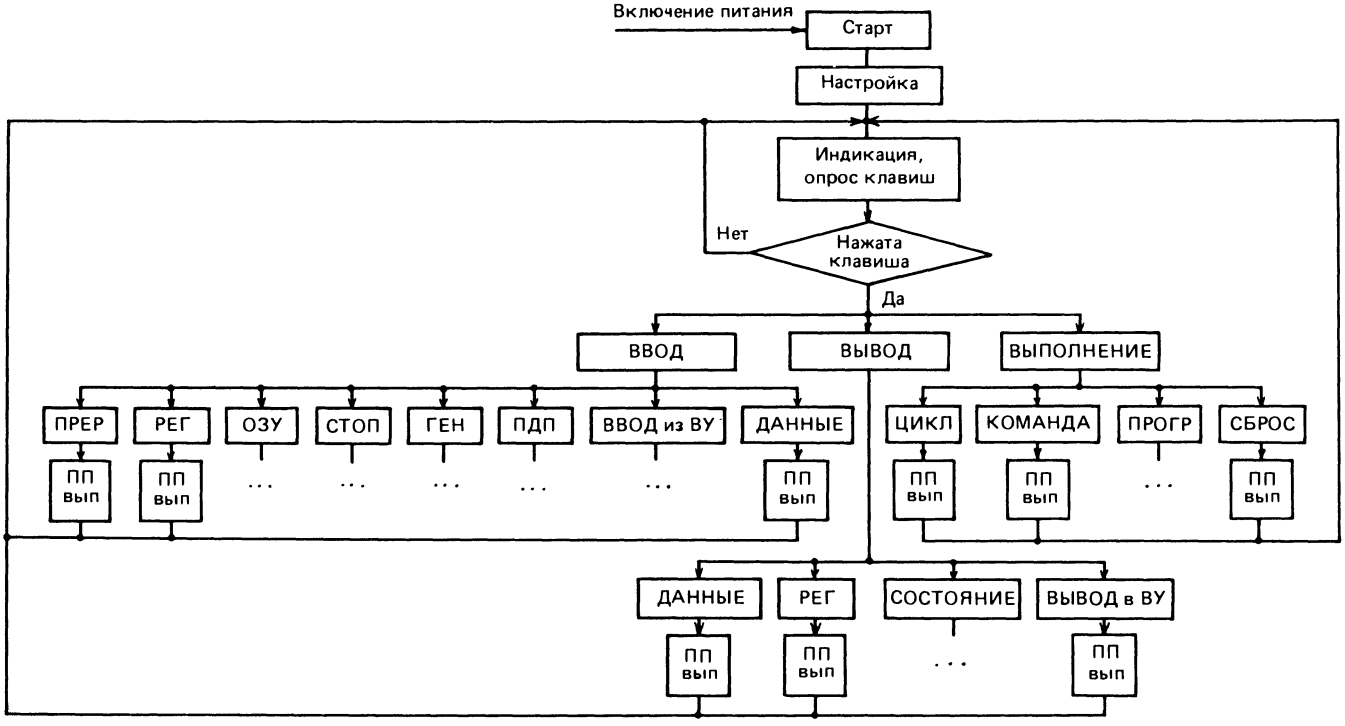


Рис. 1.15. Схема программы-монитора двухпроцессорного ОУ

МПУ. Связь эмулятора с МПУ осуществляется, таким образом, через магистраль МП (ОЭВМ). Электрическое соединение эмулятора с прототипом МПУ выполняется с помощью кабеля, оканчивающегося специальной вилкой, устанавливаемой в панельку МП (ОЭВМ) в МПУ.

Рассмотрим в качестве примера состав и основные функции программы-монитора двухпроцессорного ОУ.

На рис. 1.15 приведена схема типовой программы-монитора двухпроцессорного ОУ. После включения питания ОУ выполняется стартовая программа, производящая настройку, т. е. приведение рабочих регистров МП, счетчика команд программы и памяти эмулятора в исходное – обычно нулевое – состояние. После окончания процесса настройки управление передается программе индикации, выводящей на дисплей начальное состояние ША, ШД и состояние МП (ОЭВМ) эмулятора. Программа индикации производит также непрерывный опрос сигнала нажатой клавиши клавиатуры.

Клавиатура в рассматриваемом примере содержит 20 клавиш, предназначенных для ввода адресов, данных и управляющих воздействий: шестнадцать клавиш цифровых от 0 до F, три клавиши управления режимами работы и одна служебная клавиша. Архитектурой ОУ предусмотрено три режима работ: ВВОД, ВЫВОД и ВЫПОЛНЕНИЕ.

Режим ВВОД предназначен для ввода информации в ОУ. При этом информация может носить характер данных или кодов команд, загружаемых в память ОУ с клавиатуры или внешнего устройства, либо описывать конфигурацию отдельных компонент системы ОУ–МПУ.

Режим ВЫВОД предназначен для вывода информации на индикатор или внешнее устройство. При этом информация может отражать текущее состояние МП или содержимое памяти ОУ.

Режим ВЫПОЛНЕНИЕ предназначен для инициирования выполнения целевым МП конкретных действий, характер которых определяется самим пользователем. В соответствии с намеченным планом отладки МПУ пользователь может задать выполнение одного цикла команды, одной команды программы или некоторого фрагмента программы или же выполнить аппаратный сброс.

В каждом режиме может быть выполнен ряд команд, перечисленных ниже. Каждой команде заданного режима соответствует некоторая цифровая клавиша. Таким образом, цифровые клавиши используются как для ввода данных, так и для ввода команд.

Выполнение выбранного режима или команды инициируется нажатием служебной клавиши ВР (выбор режима).

После нажатия любой клавиши производится дешифрация ее кода и определение начального адреса подпрограммы выполнения, а затем выполнение этой подпрограммы (на рис. 1.15 выполнение подпрограммы обозначено ППвып).

При этом предварительно определяется состояние клавиш управления режимами работы ОУ: ВВОД, ВЫВОД и ВЫПОЛНЕНИЕ.

В режиме ВВОД выполняются следующие команды монитора:

РЕГ – ввод состояния программно-доступных регистров МП (ОЭВМ). Адрес (номер) рабочего регистра предварительно выводится на дисплей;

ДАННЫЕ – ввод программы и данных в память эмулятора. Начальный адрес памяти предварительно указывается на дисплее;

СТОП – ввод адресов контрольных точек останова программы пользователя (адрес и состояние МП);

ГЕН – переключение тактового генератора (внутренний или внешний);

ПДП – разрешение или запрещение прямого доступа в память эмулятора;

ВВОД из ВУ – ввод данных из внешних устройств или внешней ЭВМ. Тип ВУ и связанного с ним протокола обмена указывается на экране дисплея;

ОЗУ – указание конфигурации памяти эмулятора (начальный и конечный адрес);

ПРЕР – разрешение или запрещение прерывания МП (ОЭВМ) от внешнего источника.

В режиме ВЫВОД выполняются следующие команды монитора:

РЕГ – вывод состояния программно-доступных регистров МП (ОЭВМ) на дисплей. Адрес (номер) рабочего регистра предварительно указывается на дисплее;

ДАННЫЕ – вывод на индикацию содержимого ячейки памяти эмулятора по адресу, предварительно указанному на дисплее;

СОСТОЯНИЕ – вывод на индикацию состояния МП (ОЭВМ), т. е. информации на шинах адреса, данных и состояния МП;

ВЫВОД В ВУ – вывод данных из памяти пользователя на внешние устройства или внешнюю ЭВМ. Тип ВУ указывается на экране дисплея.

В режиме ВЫПОЛНЕНИЕ возможно выполнение следующих команд:

ЦИКЛ – выполнение одного машинного цикла текущей команды МП (ОЭВМ) и вывод на дисплей информации о последующем состоянии МП;

КОМАНДА – выполнение одной команды отлаживаемой программы, адрес которой предварительно был указан на дисплее. При последующем нажатии клавиши производится наращивание адреса и выполнение следующей команды;

ПРОГР – выполнение программы пользователя в непрерывном режиме, начиная с начального адреса, указанного на дисплее. Адрес останова может быть задан в режиме ВВОД (см. выполнение команды СТОП). В этом случае после достижения адреса останова программа прерывается и на дисплей выводится состояние МП (ОЭВМ) в точке останова;

СБРОС – приведение МП (ОЭВМ) в исходное состояние. Выполнение данной команды аналогично выполнению стартовой программы после включения питания ОУ.

После выполнения соответствующих подпрограмм производится возврат в программу индикации, далее процесс циклически повторяется.

Рассмотрим архитектуру однопроцессорных ОУ. В устройствах такого класса целевой МП (ОЭВМ) кроме функции эмуляции выполняет еще и функцию контроллера.

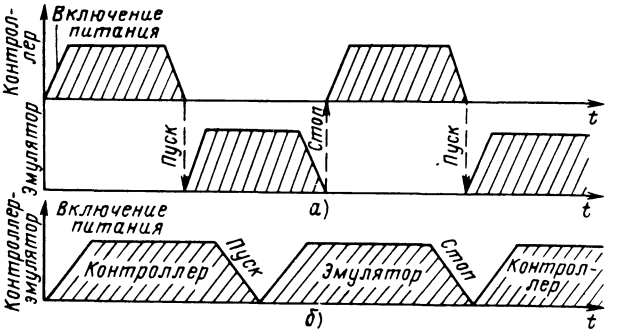


Рис. 1.16. Временна́я диаграмма работы контроллера и эмулятора в двухпроцессорных (*а*) и контроллера-эмулятора в однопроцессорных (*б*) ОУ

Временны́е диаграммы работы эмулятора и контроллера, входящих в состав любого типа ОУ, представлены на рис. 1.16. Заштрихованные области на временны́х диаграммах соответствуют активной работе указанных устройств, стрелками условно показана передача управления от одного активного устройства к другому при выполнении определенных условий. После включения питания ОУ управление передается контроллеру, затем после выполнения стартовой программы и получения условия ПУСК – эмулятору. Активная работа эмулятора прерывается после получения условия СТОП, в качестве которого могут использоваться события: КТ (контрольная точка), ШАГ (пошаговое выполнение), выполнение соответствующей команды или другие внешние условия; после этого управление вновь передается контроллеру. Выполнение программы контроллера вновь прерывается после получения условия ПУСК и т. д., процесс циклически повторяется.

Несовпадение во времени работы активных устройств (контроллера и эмулятора) и позволило объединить их функции в одном устройстве – контроллере-эмуляторе, являющемся основой структуры однопроцессорных ОУ. Временна́я диаграмма работы контроллера-эмулятора в составе однопроцессорного ОУ приведена на рис. 1.16, *б*.

Из рассмотрения рис. 1.16 становится ясным принципиальное различие в работе одно- и двухпроцессорных ОУ: в двухпроцессорных ОУ производится переключение активной работы двух различных функциональных устройств (контроллера и эмулятора), в то время как в однопроцессорном ОУ производится переключение функций одного и того же устройства – контроллера-эмулятора. Совмещение функции контроллера и эмулятора в реальных ОУ приводит к экономии до 40 % оборудования по сравнению с двухпроцессорным ОУ. Однако при этом возникают некоторые проблемы и недостатки:

необходимость практически полной переработки оборудования при переходе от одного типа целевого МП к другому;

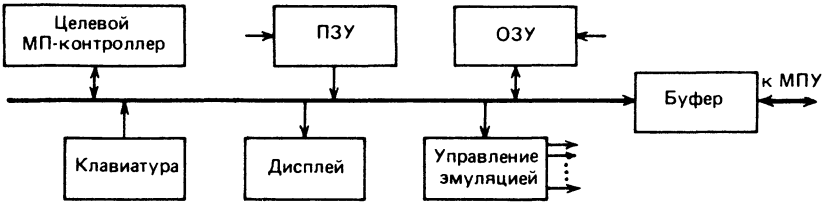


Рис. 1.17. Структурная схема однопроцессорного ОУ

ограничения, накладываемые на технические характеристики целевого МП в части использования памяти, портов ввода-вывода, рабочих регистров, системы команд и т. п.;

ограниченные возможности программы-монитора для малоразрядных целевых МП и в особенности малоразрядных ОЭВМ.

Структурная схема однопроцессорного ОУ приведена на рис. 1.17. В состав однопроцессорного ОУ входят следующие основные функциональные узлы: целевой МП-контроллер, ОЗУ и ПЗУ, клавиатура, дисплей, буфер-формирователь и узел управления эмуляцией.

Назначение указанных узлов однопроцессорного ОУ практически то же самое, что и двухпроцессорных ОУ, поэтому более подробно остановимся лишь на их отличительных особенностях.

В однопроцессорных ОУ одной из проблем является размещение программы-монитора. Как было указано выше, для хранения программы-монитора в ОУ используется ПЗУ, а для хранения программы-пользователя – ОЗУ. Известны два способа размещения монитора:

1) программа-монитор занимает часть адресного пространства памяти пользователя. Данный способ наиболее часто используется в эмуляторах высокопроизводительных 16-разрядных МП, имеющих большой диапазон адресного пространства памяти. Например, МП типа К1810ВМ86 имеет адресное пространство памяти 2^{20} байт (20-разрядная шина адреса), поэтому программа-монитор объемом 4–10К байт занимает не более 0,5 % всего адресного пространства.

С целью устранения даже этих малых потерь адресного пространства в некоторых архитектурах однопроцессорных ОУ программа-монитор реализуется в виде перемещаемой программы, что позволяет размещать ее в областях памяти, не занятых программой пользователя;

2) программа-монитор не занимает адресное пространство памяти, а является фоновой или маскируемой. Данный способ используется в эмуляторах малоразрядных (4–8 бит) МП (ОЭВМ), в которых из-за ограниченности адресного пространства памяти, сравнимого по величине с объемом программы-монитора, такие потери являются недопустимыми. Фоновое ПЗУ размещается в тех же адресах, что и память пользователя, поэтому для управления обращением к ним необходимы дополнительные аппаратные средства, позволяющие стробировать выборку либо из фоно-

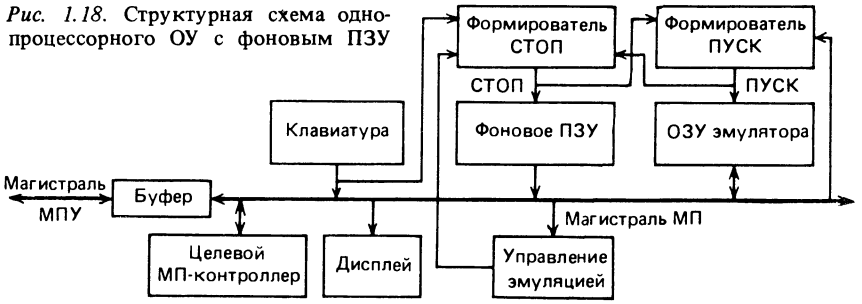


Рис. 1.18. Структурная схема однопроцессорного ОУ с фоновым ПЗУ

вого ПЗУ, либо из ОЗУ эмулятора. Рассмотрим один из способов передачи управления в однопроцессорном ОУ при наличии фонового ПЗУ, схема которого приведена на рис. 1.18. В состав ОУ входят: целевой МП-контроллер, фоновое ПЗУ, ОЗУ эмулятора, дисплей, клавиатура, буфер магистрали МПУ, узел управления эмуляцией и два формирователя сигналов СТОП и ПУСК.

Сигнал СТОП формируется при нажатии на клавиатуре любой из клавиш, выдающей команды монитора, либо при поступлении сигнала из узла управления эмуляцией, а сигнал ПУСК – при появлении на магистрали МП определенной кодовой комбинации, на которую настроен соответствующий формирователь. При выдаче сигнала СТОП происходит одновременно сброс сигнала ПУСК и, напротив, при выдаче сигнала ПУСК – сброс сигнала СТОП.

После выдачи сигнала СТОП управление передается программе-монитору, хранящейся в фоновом ПЗУ. После выполнения подпрограммы команд монитора на магистрали МП программно формируется кодовая комбинация, на которую настроен формирователь сигнала ПУСК, и управление передается программе пользователя.

Отладочные комплексы. Отладочные комплексы предназначаются для отладки аппаратурных и программных средств сложных МПУ, объем управляющих программ которых может быть значительно больше 4 К байт.

Отладочные комплексы строятся на базе ЭВМ с набором периферийных устройств (дисплей, клавиатура, накопители на гибких и жестких магнитных дисках, печатающие устройства и т. п.), что, естественно, приводит к увеличению их стоимости, габаритов, веса и потребляемой энергии, но, с другой стороны, значительно расширяет их функциональные возможности.

По сравнению с отладочными устройствами отладочные комплексы имеют много преимуществ, из которых отметим два основных:

1) возможность использования при разработке и отладке программного обеспечения МПУ развитых систем программирования МП (ОЭВМ) с набором языков высокого уровня, редакторами текста и связей, программно-логическими моделями и набором управляющих программ

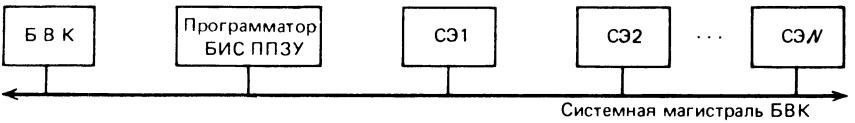


Рис. 1.19. Схема универсального отладочного комплекса

эмуляторов, что значительно повышает производительность труда программистов. Известно, например, что только использование языков высокого уровня в 8–10 раз повышает производительность труда программистов по сравнению с использованием машинно-ориентированных языков;

2) возможность автоматизации процессов документирования и отображения оперативной информации при отладке, долговременное хранение и быстрая выдача по запросу больших массивов информации с магнитных носителей (диски, ленты), что также значительно повышает производительность труда.

Отладочные комплексы можно разделить на две группы: универсальные и специализированные. Универсальные отладочные комплексы предназначены для проектирования и отладки МПУ на базе нескольких различных типов или семейств МП (ОЭВМ). Специализированные отладочные комплексы предназначены для проектирования и отладки МПУ на базе одного типа или семейства МП (ОЭВМ).

В последнее время универсальные отладочные комплексы получили большее распространение, чем специализированные, поэтому при дальнейшем изложении подробно рассмотрим архитектуру именно этой группы отладочных средств. На рис. 1.19 приведена схема универсального отладочного комплекса (в дальнейшем изложении отладочный комплекс).

В общем случае в состав отладочного комплекса входят: базовая ЭВМ или базовый вычислительный комплекс (БВК) со штатными периферийными устройствами; набор из нескольких схемных эмуляторов СЭ используемых типов целевых МП (ОЭВМ) и программатор БИС ППЗУ, подключаемые к системной магистрали БВК.

Базовая ЭВМ (БВК) в составе отладочного комплекса выполняет те же функции, что и контроллер в составе отладочных устройств плюс дополнительные функции автоматизации программирования МП (ОЭВМ), документирования, долговременного хранения информации и т. п.

Функции схемных эмуляторов отладочного комплекса в принципе не отличаются от функций эмулятора, входящего в состав отладочных устройств. То же можно сказать и в отношении программатора БИС ППЗУ.

Как указывалось выше, универсальные отладочные комплексы можно разделить на два класса: отладочные комплексы широкого применения (средней производительности) и отладочные комплексы высокой производительности.

Рассмотрим подробнее архитектуру этих классов отладочных средств.

Отладочные комплексы широкого применения являются, как следует уже из их названия, массовыми отладочными средствами и строятся на базе персональных ЭВМ (ПЭВМ). Персональные ЭВМ как класс вычислительных средств появились относительно недавно, но уже получили очень широкое распространение. В мире насчитывается несколько десятков миллионов штук ПЭВМ, находящихся в эксплуатации. Основные области применения ПЭВМ – автоматизация индивидуального труда при решении информационно-расчетных и управленческих задач. Однако ПЭВМ могут использоваться и для целей управления оборудованием, хотя здесь их возможности значительно меньше, чем возможности традиционных ЭВМ (микроЭВМ).

Схема отладочного комплекса широкого применения приведена на рис. 1.20. В состав отладочного комплекса входят следующие основные функциональные устройства: ПЭВМ; универсальный схемный эмулятор УСЭ; блок целевого микропроцессора БЦМП.

Персональная ЭВМ содержит в своем составе: микроЭВМ с контроллерами периферийных устройств и интерфейсом ввода-вывода, накопитель на гибком магнитном диске НГМД, алфавитно-цифровое печатающее устройство АЦПУ, дисплей, клавиатуру.

Указанный состав оборудования имеют большинство моделей современных отечественных и зарубежных профессиональных ПЭВМ (ЕС1840, ДВК-2М, ДВК-3, IBM PC и др.). Некоторые модели высокопроизводительных ПЭВМ имеют в своем составе также малогабаритные накопители на жестких магнитных дисках винчестерского типа („Электроника 85", ЕС1841, IBM PC XT, IBM PC AT и др.).

МикроЭВМ, выполняющая в составе ПЭВМ функции центрального процессора, мало отличается от управляющих микроЭВМ по своему быстродействию и системе команд. То же можно сказать и в отношении характеристик используемых в составе ПЭВМ периферийных устройств.

Основные отличия ПЭВМ от управляющих ЭВМ заключается в имеющихся каналах ввода-вывода, а также операционной системе, системе программирования и в особенности – в пакетах прикладных программ.

Одной из основных особенностей ПЭВМ при использовании их в качестве управляющих ЭВМ является наличие только низкосортного последовательного канала ввода-вывода типа "Стык 2" (С2) по ГОСТ 23675–79. Данный канал может использоваться в качестве системного канала для подключения дополнительного оборудования. Скорости обмена информацией по каналу С2 составляют 75–19 200 бит/с, что на несколько порядков ниже, чем скорости обмена информацией по каналам ввода-вывода управляющих ЭВМ. Канал С2 находит применение при асинхронной или синхронной передаче данных в сетях ЭВМ, для управления низкоскоростным периферийным оборудованием (печатающее устройство, дисплей) и т. п. В минимальном варианте канал С2

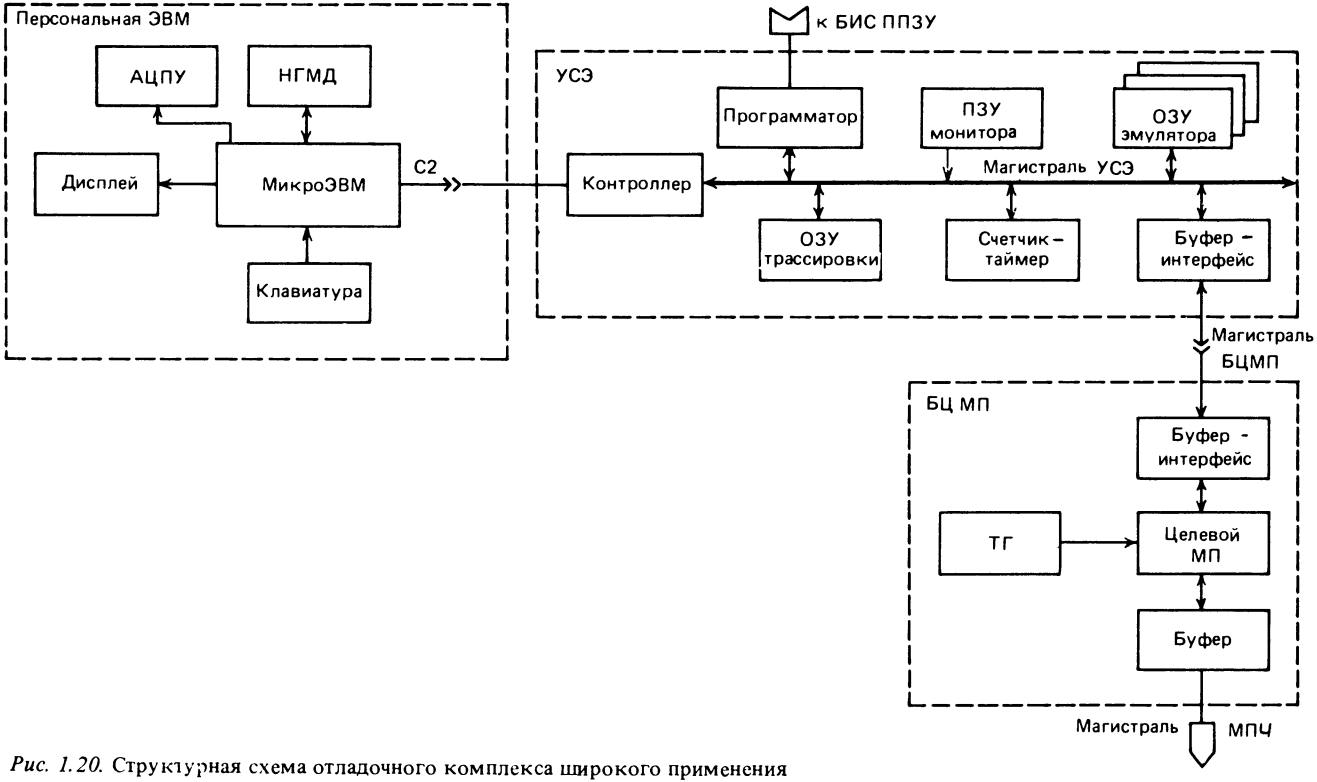


Рис. 1.20. Структурная схема отладочного комплекса широкого применения

включает в свой состав 6 одноразрядных шин, в максимальном варианте – 48 одноразрядных шин. Данные передаются по каналу С2 в обе стороны (от конечного оборудования данных к аппаратуре передачи данных и обратно) в последовательном коде.

При использовании данного канала в качестве системного в отладочном комплексе ПЭВМ выполняет функции интеллектуального терминала, автоматизации разработки и отладки программ с помощью кросс-систем программирования, а также управления режимами эмуляции. Скоростные технические характеристики канала С2 оказываются достаточными для построения однопользовательских отладочных комплексов для отладки однопроцессорных МПУ. Ограничения канала С2 проявляются в этом случае лишь при использовании в составе эмулятора памяти пользователя большого объема (более 1М байт), однако данный случай является довольно редким.

При достаточно большой номенклатуре поддерживаемых целевых МП (более пяти типов) оказывается экономически целесообразным построить схемный эмулятор в виде неизменяемой (универсальной) части, не зависящей от типа применяемого МП, и изменяемой части, зависящей от типа целевого МП.

Неизменяемую часть схемного эмулятора называют универсальным схемным эмулятором (УСЭ), а изменяемую – блоком целевого МП (БЦМП). На рис. 1.20 в состав УСЭ входят: контроллер, ПЗУ монитора, ОЗУ эмулятора ОЗУ трассировки, счетчик-таймер, программатор БИС ППЗУ, буфер-интерфейс магистрали БЦМП.

В состав БЦМП входят: целевой МП, тактовый генератор ТГ, буфер-интерфейс магистрали БЦМП, буфер магистрали МПУ.

Контроллер УСЭ выполняет следующие функции: сопряжение с каналом С2 ПЭВМ; управление обменом информацией между ПЭВМ и ОЗУ пользователя, ОЗУ трассировки и счетчиком-таймером; управление режимами эмуляции УСЭ; управление программатором БИС ППЗУ.

В реальных структурах УСЭ контроллер реализуется обычно на базе 8- и 16-разрядных МП типа К580ВМ80, 8085, Z80 или К1810ВМ86.

С целью обеспечения применения различных типов целевых МП (ОЭВМ) и получения при этом минимальной аппаратной избыточности в УСЭ ОЗУ эмулятора строится по модульному принципу и его объем может наращиваться от минимальной до максимальной величины для данного типа МП. На практике в составе УСЭ объем ОЗУ пользователя может изменяться в пределах 4–256К байт. В некоторых типах УСЭ может изменяться также разрядность шины данных ОЗУ пользователя (8 или 16 бит), что достигается либо программно, либо с помощью сменных перемычек на плате ОЗУ.

Оперативное ЗУ трассировки предназначено для хранения трассы ("следа") МП при выполнении программ пользователя. Объем ОЗУ трассировки в реальных УСЭ изменяется в пределах от 256 до 4К слов, разрядность – от 32 до 48 бит. Оперативное ЗУ трассировки строится на

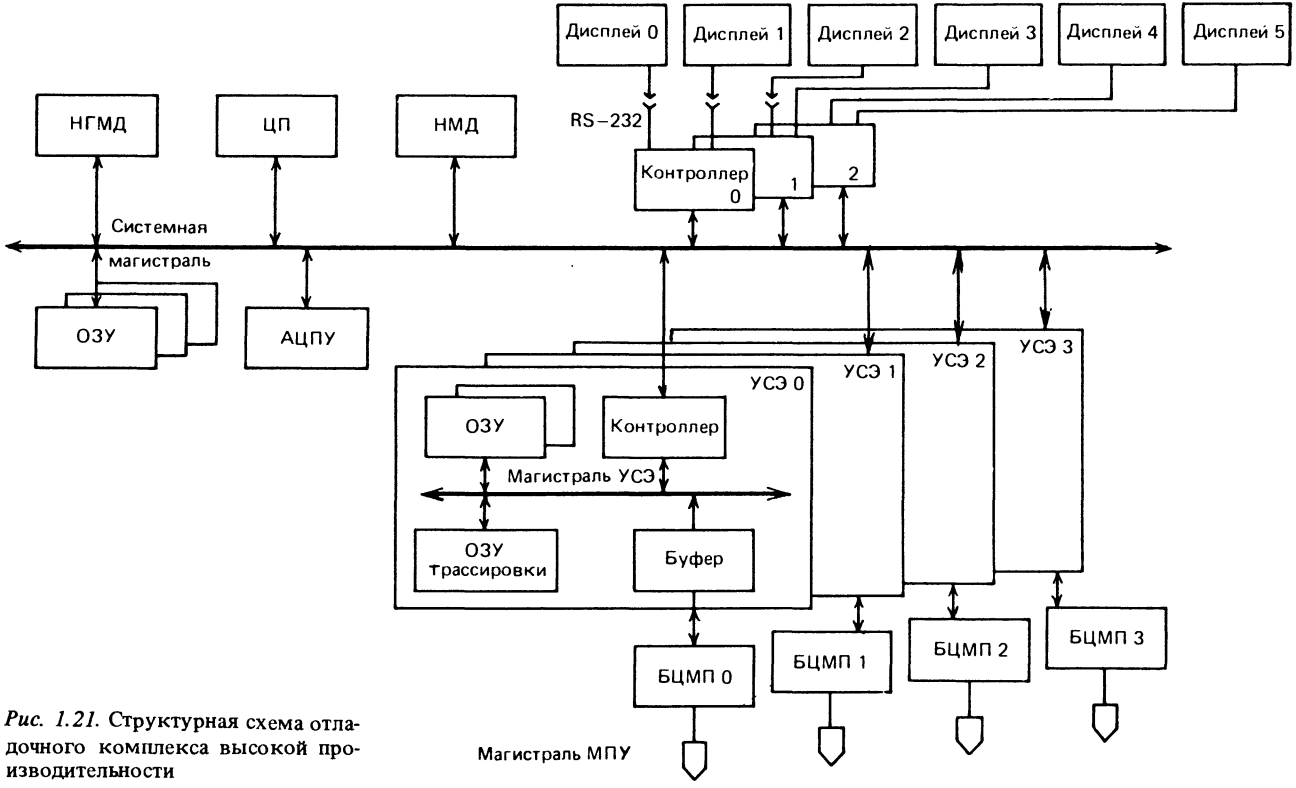


Рис. 1.21. Структурная схема отладочного комплекса высокой производительности

быстродействующих биполярных БИС ОЗУ, чтобы не замедлять режим прогона программ пользователя в реальном времени. В эмуляторах ОЗУ трассировки организуется в виде памяти магазинного типа (первый вошел – первый вышел), и его содержимое считывается в ПЭВМ после остановки по контрольной точке для анализа и поиска неисправностей. Максимальная разрядность ОЗУ трассировки выбирается исходя из разрядности шин адреса данных и управления целевого МП.

Счетчик-таймер предназначается для измерения времени выполнения отдельных программ или подпрограмм пользователя.

Постоянное ЗУ используется для хранения программы загрузки памяти УСЭ и программ тестирования УСЭ.

Блок ЦМП конструктивно выполняется сменным, подключается к УСЭ с помощью гибкого кабеля с разъемом. Связь БЦМП с отлаживаемым МПУ осуществляется с помощью кабеля со специальной вилкой, вставляемой в панельку МП на плате МПУ.

В описанной структуре отладочного комплекса при переходе от одного типа целевого МП к другому необходимо изменить: БЦМП (отстыковывается от УСЭ и заменяется нужным типом); объем памяти пользователя (в случае необходимости); разрядность ШД (в случае необходимости); кросс-систему программирования целевого МП (ОЭВМ) и управляющую программу УСЭ – в ПЭВМ.

На рис. 1.21 приведена схема отладочного комплекса высокой производительности.

В отличие от описанного выше отладочного комплекса средней производительности данный класс комплексов строится на базе высокопроизводительных универсальных или специализированных микроЭВМ с набором периферийных устройств. Особенностями данного типа комплексов является обязательное наличие в их составе НМД большого объема (более 20М байт), а также высокоскоростного параллельного системного канала для подключения дополнительного оборудования (контроллеров дисплейных терминалов, универсальных схемных эмуляторов и т.п.), которое необходимо для организации многопользовательского режима отладки многопроцессорных устройств.

В состав отладочного комплекса высокой производительности в рассматриваемом примере входят: высокопроизводительный ЦП с быстродействием более 1 млн. операций/с; модульное ОЗУ большого объема (более 1М байт); НГМД; НМД объемом не менее 20М байт; высокопроизводительное АЦПУ; 6 дисплейных терминалов с контроллерами; 4 универсальных схемных эмулятора (УСЭО–УСЭ3); 4 сменных БЦМП 0–3.

Комплекс позволяет производить отладку многопроцессорных МПУ (до четырех МП одновременно) и одновременно работать шести пользователям. Операционная система управляющей микроЭВМ должна позволять многопользовательский и многопрограммный режимы работы. В качестве таких операционных систем в современных отладочных комп-

лексах высокой производительности наиболее часто используется UNIX или ее модификации.

Структура и основные функции УСЭ и БЦМП примерно аналогичны структуре и основным функциям описанных выше аналогичных устройств за исключением того, что связь УСЭ с ЦП осуществляется через высокоскоростную системную шину.

Входящие в состав комплекса три контроллера дисплейных терминалов позволяют одновременно работать шести пользователям. Связь дисплейных терминалов с контроллерами осуществляется через канал С2 (аналог RS-232), описанный выше.

ГЛАВА 2

УНИВЕРСАЛЬНЫЙ ОТЛАДОЧНЫЙ КОМПЛЕКС

2.1. Архитектура, технические параметры и основные функциональные возможности

Рассмотренные ранее принципы организации универсального отладочного комплекса (УОК) были реализованы при создании вычислительного отладочного комплекса МИКРОСОТ (микропроцессорная система отладки). МИКРОСОТ может быть классифицирован как персональный УОК широкого применения и средней производительности и рассчитан на отладку однопроцессорных МПУ.

МИКРОСОТ предназначен для автоматизации процессов проектирования, отладки и испытаний программных и аппаратных средств МПУ. МИКРОСОТ рассчитан на применение при отладке МПУ, построенных на базе различных типов МП и ОЭВМ. Функционально МИКРОСОТ включает в свой состав базовый вычислительный комплекс, набор блоков эмуляторов, программатор и набор пакетов комплексов отладочных программ (КОП). Типы используемых блока эмулятора и КОП однозначно определяются типом МП или ОЭВМ отлаживаемого МПУ. В зависимости от типа блока эмулятора, включаемого в состав комплекса, МИКРОСОТ позволяет автоматизировать разработку программных и аппаратных средств МПУ, построенных на базе МП серий КР580ВМ80, КР1810ВМ86 и ОЭВМ КР1816ВЕ48.

Комплекс также может быть использован в качестве инструментального для разработки программных средств МПУ, построенных на базе БИС микропроцессорных комплектов серий К581, К588, КР1801 и других, совместимых по системе команд с ДВК-2М и микроЭВМ „Электроника 60".

МИКРОСОТ представляет собой комплект программно-аппаратурных средств на основе диалогового вычислительного комплекса ДВК-2М. В основу организации МИКРОСОТ положен принцип объединения функционально и конструктивно законченных аппаратурных модулей, каж-

Рис. 2.1. Общий вид МИКРОСОТ

дый из которых имеет необходимую программную поддержку в виде комплекса отладочных программ.

Общий вид МИКРОСОТ представлен на рис. 2.1. В состав МИКРОСОТ входят: диалоговый вычислительный комплекс ДВК-2М; блоки эмуляторов БЭ-580; БЭ-816 и БЭ-810; комплексы отладочных программ КОП-580, КОП-816, КОП-810; модульный программатор БИС ППЗУ „Электроника МС9401'; коннектор; кабель соединительный.

Комплекс поставляют как в полном составе, так и в конфигурации, определяемой пользователем в соответствии с его потребностями.

Функционально-модульный принцип построения МИКРОСОТ допускает расширение состава его программных и аппаратурных средств путем включения дополнительных блоков эмуляторов и соответствующих КОП для вновь разрабатываемых и перспективных типов МП и ОЭВМ.

Взаимное соединение составных частей комплекса осуществляется в соответствии со схемой соединений, представленной на рис. 2.2. Различные схемные эмуляторы и программатор подключаются к системной магистрали БВК поочередно. Необходимость в переключении аппаратурных средств возникает сравнительно редко, а затраты времени минимальны — достаточно переключить соединительный кабель от одного блока к другому.

Базовый вычислительный комплекс обеспечивает управление блоками эмуляторов и программатором, автоматизирует процесс подготовки, трансляции и отладки программ пользователя, обрабатывает и представляет пользователю в удобном для восприятия виде информацию о поведении отлаживаемой системы, обеспечивает документирование процесса отладки, осуществляет диалог с оператором.

Каждый блок эмулятора представляет собой физическую модель МП или ОЭВМ и памяти разрабатываемого МПУ. Назначение, функциональ-

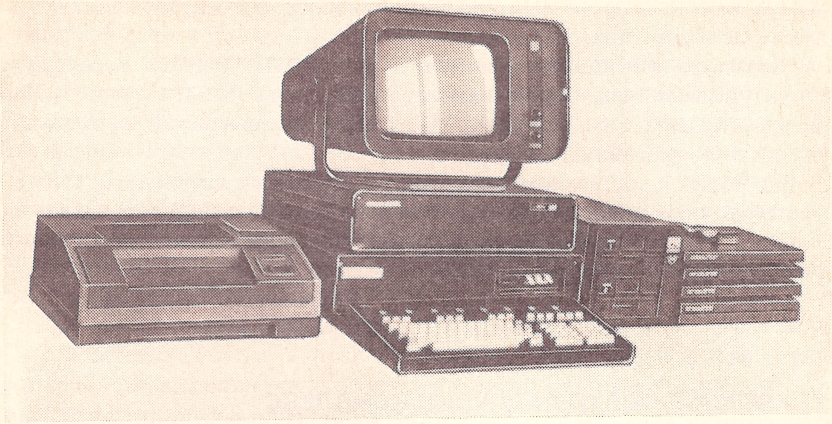


Рис. 2.1. Общий вид МИКРОСОТ

дый из которых имеет необходимую программную поддержку в виде комплекса отладочных программ.

Общий вид МИКРОСОТ представлен на рис. 2.1. В состав МИКРОСОТ входят: диалоговый вычислительный комплекс ДВК-2М; блоки эмуляторов БЭ-580; БЭ-816 и БЭ-810; комплексы отладочных программ КОП-580, КОП-816, КОП-810; модульный программатор БИС ППЗУ „Электроника МС9401' ; коннектор; кабель соединительный.

Комплекс поставляют как в полном составе, так и в конфигурации, определяемой пользователем в соответствии с его потребностями.

Функционально-модульный принцип построения МИКРОСОТ допускает расширение состава его программных и аппаратурных средств путем включения дополнительных блоков эмуляторов и соответствующих КОП для вновь разрабатываемых и перспективных типов МП и ОЭВМ.

Взаимное соединение составных частей комплекса осуществляется в соответствии со схемой соединений, представленной на рис. 2.2. Различные схемные эмуляторы и программатор подключаются к системной магистрали БВК поочередно. Необходимость в переключении аппаратурных средств возникает сравнительно редко, а затраты времени минимальны — достаточно переключить соединительный кабель от одного блока к другому.

Базовый вычислительный комплекс обеспечивает управление блоками эмуляторов и программатором, автоматизирует процесс подготовки, трансляции и отладки программ пользователя, обрабатывает и представляет пользователю в удобном для восприятия виде информацию о поведении отлаживаемой системы, обеспечивает документирование процесса отладки, осуществляет диалог с оператором.

Каждый блок эмулятора представляет собой физическую модель МП или ОЭВМ и памяти разрабатываемого МПУ. Назначение, функциональ-

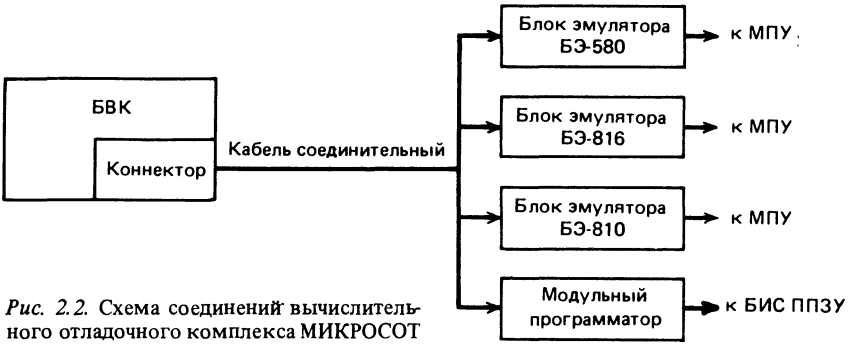


Рис. 2.2. Схема соединений вычислительного отладочного комплекса МИКРОСОТ

ные возможности и структура основных функциональных узлов блоков эмуляторов в общем виде были изложены при рассмотрении метода внутрисхемной эмуляции.

Модульный программатор предназначен для формирования под управлением БВК электрических сигналов, обеспечивающих программирование и контроль памяти МПУ на завершающем этапе отладки. Учитывая универсальный характер отладочного комплекса, программатор должен обеспечить обслуживание по возможности максимально полной номенклатуры БИС программируемых ПЗУ, в том числе: СППЗУ серий К573, ППЗУ серий КР556 и К155, встроенной программируемой памяти целевых ОЭВМ, а также программируемых логических матриц. Модульная структура программатора обеспечивает его оперативную адаптацию для вновь разрабатываемых БИС ППЗУ [14].

Пакеты КОП предназначены для программной поддержки проектирования и отладки АС и ПС МПУ, интеграции АС и ПС комплекса, а также комплексной отладки МПУ.

Минимально необходимый состав каждого пакета КОП включает: кросс-ассемблер целевого МП, кросс-редактор связей, программно-логическую модель, управляющие программы блока эмулятора целевого МП и программатора. Пакет КОП предназначен для работы в среде операционной системы БВК, поэтому последняя также включается в состав программного обеспечения комплекса.

Функциональные возможности МИКРОСОТ обеспечивают проведение следующих четырех основных этапов разработки МПУ:

1) разработку текста программы МПУ и получение загрузочного модуля: формирование и корректировку исходного текста прикладной программы разрабатываемого МПУ на языке АССЕМБЛЕР с помощью редактора текста и запись ее на магнитный диск; трансляция текста исходных программных модулей с помощью кросс-ассемблера и получение распечатки результатов трансляции на печатающем устройстве БВК, а также вывод результатов трансляции на дисплей БВК и получение файлов перемещаемых программных модулей в объектных кодах на магнит-

ном диске; вызов и использование библиотеки стандартных программ; компоновку отдельных объектных модулей (файлов) в единый загрузочный модуль и размещение его в заданном пользователем адресном пространстве памяти разрабатываемого МПУ с помощью редактора связей; формирование единого загрузочного файла на магнитном диске. Реализация перечисленных функциональных возможностей осуществляется с применением БВК под управлением комплекса отладочных программ;

2) отладку программ пользователя: индикация и модификация содержимого любой ячейки памяти отлаживаемой программы; получение твердой копии содержимого любого участка памяти отлаживаемой программы; прогон всей программы или любой ее части; трассировка; имитация распределения памяти отлаживаемой программы на ПЗУ и ОЗУ (функция картирования памяти); имитация средств ввода-вывода и внешних прерываний; останов отлаживаемой программы при обращении по заданному адресу; индикация и модификация содержимого регистров моделируемого МП [8]. Отладка осуществляется на программно-логической модели не в реальном масштабе времени с применением БВК. На первых двух этапах модульный программатор и блоки эмуляторов не используются;

3) диагностирование и комплексную отладку программных и аппаратурных компонент МПУ: эмуляцию МП и памяти МПУ; выполнение программ пользователя в реальном масштабе времени с остановом на задаваемых контрольных точках, а также в пошаговом или поцикловом режимах; запуск программ пользователя с произвольного или текущего адреса; останов МП по команде оператора; индикацию и модификацию содержимого памяти эмулятора или МПУ, а также счетчика команд, программно доступных регистров общего назначения и флагов состояния процессора; использование внутренних или внешних цепей синхронизации (тактового генератора); программное изменение конфигурации памяти МПУ и блока эмулятора; дизассемблирование загрузочных модулей программы (перевод модулей программы из машинных кодов в символическое представление). Отладка на этом этапе производится в реальном масштабе времени с использованием БВК, соответствующего микропроцессору МПУ внутрисхемного эмулятора и управляющей программы, входящей в состав комплекса отладочных программ;

4) программирование БИС ППЗУ: настройку программных и аппаратурных средств комплекса на заданный тип БИС ППЗУ; загрузку в буферное ОЗУ БВК заданного файла загрузочного модуля с магнитного диска; запись кодов ППЗУ в буферное ОЗУ БВК с клавиатуры БВК; запись кодов из буферного ОЗУ БВК во внешнее ЗУ на гибком магнитном диске; считывание данных из БИС ППЗУ в буферное ОЗУ БВК; запись в БИС ППЗУ содержимого буферного ОЗУ БВК (режим программирования); сравнение содержимого буферного ОЗУ БВК и БИС

ППЗУ и вывод на дисплей сообщения о результате сравнения; вывод на дисплей и печать содержимого буферного ОЗУ БВК.

Ниже приведены основные эксплуатационные характеристики комплекса:

Площадь, $м^2$, не более 1,5
Потребляемая мощность, ВТ, не более 650
Масса устройств МИКРОСОТ, кг, не более:
ДВК-2М 112
БЭ-580 1,9
БЭ-816 1,1
БЭ-810 2,0
программатор 2,2

2.2. Базовый вычислительный комплекс

Общие требования к базовому вычислительному комплексу. Выбор базового вычислительного комплекса в значительной мере определяет такие важные параметры и характеристики УОК, как состав отладочного программного (крос-программного) обеспечения, функциональные возможности по отладке аппаратурной и программной компонент МПУ, удобство поддержания диалога оператор – комплекс, стоимость, эргономические характеристики и т. д.

Для построения универсального отладочного комплекса средней производительности необходим БВК, удовлетворяющий следующим основным требованиям: объем оперативной памяти – не менее 56 К байт; быстродействие центрального процессора при выполнении коротких операций типа регистр–регистр – не менее 200 тыс. операций/с; наличие минимального комплекта периферийных устройств в составе: алфавитно-цифровой дисплей, клавиатура, печатающее устройство, накопитель на гибких магнитных дисках; наличие канала для подключения дополнительных аппаратурных средств; массовость объема выпуска БВК и их доступность; наличие развитой операционной системы со средствами программирования; стоимость БВК – 10–20 тыс. руб.

Требование к оперативной памяти в первую очередь определяется объемом программ, входящих в состав КОП. Опыт разработки отладочных комплексов показывает, что объем управляющих программ (разработанных на языке ПАСКАЛЬ) для блоков эмуляторов может составлять 40–50К байт и более. Встречаются ситуации, когда становится актуальным вопрос о минимизации объема управляющей программы даже за счет сокращения сервисных возможностей по представлению информации о процессе отладки. Недостаточный объем оперативной памяти при работе кросс-программного обеспечения может привести к необходимости частых обращений к внешнему ЗУ на магнитном диске (при организации работы программ с перекрытием), что заметно увеличивает затраты времени при подготовке и отладке целевых программ.

Требование к быстродействию БВК связано с двумя аспектами. Во-первых, необходимо обеспечить приемлемое быстродействие комплекса при трансляции исходных текстов отлаживаемых программ (200–2000 байт/мин) и их прогоне на программно-логической модели. Во-вторых, время обработки команд оператора и выдачи результата при работе с управляющими программами блоков эмуляторов (на этапе программно-аппаратурной отладки) не должно заметно превышать времени ввода команд с клавиатуры оператором или же времени, необходимого оператору для восприятия результатов выполнения команд.

Требованиям минимально необходимого объема оперативной памяти и быстродействия удовлетворяют практически все современные ЭВМ средней производительности.

Минимально необходимый состав периферийного оборудования определяется функциями, выполняемыми отладочными комплексами средней производительности. Периферийные устройства БВК необходимы для ввода исходных текстов целевых программ и организации диалога с пользователем (клавиатура, дисплей), выдачи оперативной информации о ходе процесса отладки и его документирования (дисплей, печатающее устройство), долговременного хранения исходных текстов и подготовленных загрузочных модулей отлаживаемых программ, а также операционной системы (накопитель на гибких магнитных дисках).

Дисплей УОК, как правило, выполняется на базе ЭЛТ и объем выводимой на индикацию информации составляет 24X80 символов. Требования к клавиатуре носят в основном эргономический характер.

Основное требование к печатающему устройству – быстродействие. Это требование особо важно при документировании процесса отладки, когда время печати задерживает отражение результатов выполнения команд оператора. Печатающее устройство может быть использовано и для подготовки программной документации на завершающем этапе проектирования МПУ, объем которой достигает сотен страниц и фактор быстродействия также становится существенным.

Для управления всем оборудованием БВК необходима развитая операционная система. Операционная система вместе с кросс-системой программирования, файлами исходного текста и загрузочного модуля обычно размещается на гибких магнитных дисках, минимальный объем которых должен быть не менее 0,5М байт.

Наличие стандартного канала для подключения внутрисхемных эмуляторов и программатора освобождает от необходимости доработки БВК в этой части при комплексировании системы отладки. В совокупности с массовостью распространения наличие канала помогает пользователю эффективно эксплуатировать для целей отладки уже имеющиеся в его распоряжении БВК, дополняя его необходимыми программными и аппаратурными модулями. При оценке канала связи следует иметь в виду, что использование для этой цели системной магистрали БВК позволяет несколько сократить аппаратурные затраты.

Наличие параллельного канала связи является более предпочтительным по сравнению с последовательным каналом в силу его большей пропускной способности. Выполнение команд оператора в процессе отладки сопровождается передачей большого объема данных и управляющих воздействий между БВК и блоком эмулятора. Достижение требуемого быстродействия при обмене информации осуществляется правильным сочетанием выбора канала связи и протокола обмена, который определяется структурой внутрисхемного эмулятора.

Наконец, требования массовости объема выпуска БВК, их доступности и относительно небольшой стоимости являются для УОК широкого применения понятными, не требуют особых пояснений и могут быть решающими при выборе типа БВК.

Всем перечисленным требованиям к БВК в наибольшей степени отвечают современные персональные ЭВМ (ПЭВМ), в особенности те модели ПЭВМ, которые предназначены для профессионального использования (профессиональные ПЭВМ) на предприятиях, учреждениях, в организациях и т. п. Ниже приведены основные технические характеристики некоторых моделей отечественных профессиональных ПЭВМ, которые могут применяться в качестве БВК в составе УОК средней производительности.

Диалоговые вычислительные комплексы (ДВК). Широко распространенные диалоговые вычислительные комплексы предназначены для использования: в системах автоматизированного проектирования в качестве инструментальных ЭВМ для отработки программного обеспечения; в системах управления технологическими процессами; в вычислительных системах и сетях в качестве терминальной ЭВМ; в информационно-справочных и информационно-поисковых системах; при решении различных инженерно-технических задач.

По способу взаимодействия с пользователем ДВК можно отнести к диалоговым, по представлению ресурсов – к персональным, по ориентации на область применения – к профессиональным ЭВМ [9].

Семейство ДВК включает в свой состав несколько моделей:

ДВК-1, в состав которого входит одноплатная микроЭВМ и алфавитно-цифровой дисплей с клавиатурой; НГМД и печатающее устройство отсутствуют;

ДВК-2 и ДВК-2М, отличающиеся друг от друга типом накопителя на гибких магнитных дисках. Модель ДВК-2 включает в свой состав НГМД на гибких магнитных дисках диаметром 203 мм, а модель ДВК-2М – накопитель на гибких магнитных мини-дисках диаметром 133 мм. Обе модели включают в свой состав также дисплей и печатающее устройство;

ДВК-3М2, отличающийся от предыдущих моделей (ДВК-2, ДВК-2М) новым, более быстродействующим ЦП и алфавитно-цифровым/графическим дисплеем.

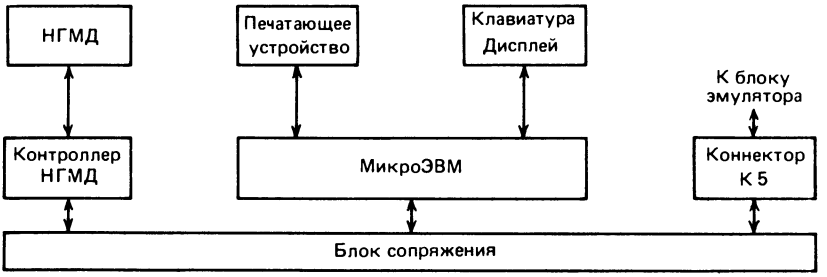


Рис. 2.3. Схема ДВК-2М

Все модели ДВК совместимы между собой снизу вверх по системе команд центрального процессора, а также по архитектуре. Все модели ДВК в качестве системного канала имеют совмещенную по адресам и данным общую шину, выполненную в соответствии со стандартом на межмодульный параллельный интерфейс (МПИ). Системный канал ДВК конструктивно и электрически соответствует каналу микроЭВМ „Электроника 60".

С системным каналом непосредственно связан ЦП и память, все остальные функциональные устройства связаны с МПИ через контроллеры.

Системный канал ДВК включает в свой состав 38 сигнальных линий: 16 линий адреса и данных (совмещенные) и 22 линии управления (входные и выходные).

Связь между устройствами, подключенными к МПИ, осуществляется по принципу активный – пассивный. В любой момент времени только одно устройство является активным и управляет циклами обмена информацией в канале. Передача данных через МПИ осуществляется по асинхронному принципу, т. е. на сигнал запроса активного устройства должен поступить ответ от пассивного устройства, поэтому процесс обмена не зависит от их быстродействия по выборке и приему данных. Обмен данными может осуществляться в программном режиме, по прерыванию и в режиме прямого доступа к памяти. Наивысший приоритет имеет устройство, электрически ближе расположенное к ЦП.

Схема модели ДВК-2М, являющейся одной из наиболее распространенных в семействе ДВК и используемой в качестве БВК в отладочном комплексе вычислительном МИКРОСОТ, приведена на рис. 2.3.

В состав ДВК-2М входят следующие основные функциональные устройства: одноплатная микроЭВМ типа „Электроника НМС 11100.1"; сдвоенный накопитель на гибком магнитном диске НГМД типа „Электроника НГМД-6022"; алфавитно-цифровой дисплей с клавиатурой типа 15ИЭ-00-013; печатающее устройство „Электроника УВВП4-30-004" или термопечатающее устройство 15ВВП-80-002; контроллер НГМД (КНГМД); блок сопряжения.

Модель ДВК-2М имеет следующие основные характеристики

(система команд совместима с системой команд микроЭВМ „Электроника 60") :

Число команд центрального процессора	64
Число методов адресации	8
Число типов команд	3
Разрядность ЦП, бит	16
Объем оперативной памяти, К байт	56
Быстродействие ЦП, тыс. операций/с	400±100
Объем постоянной памяти, К байт	8
Число регистров общего назначения	8
Габариты микроЭВМ, мм	252×296×12
Потребляемая микроЭВМ мощность, Вт	15
Объем памяти НГМД, К байт	432
Скорость печати (для ПУ типа „Электроника УВВП4-30-004"), зн/с	30
Объем выводимой на индикацию информации	24 строки × × 80 символов

Операционная система ОС ДВК является подмножеством системы РАФОС, аналогом системы ФОДОС и включает в себя: управляющую систему; файловую систему; систему программирования, редактирования и отладки, включающую редакторы EDIT, К-52, SCREEN и допускающую работу с языками программирования АССЕМБЛЕР, ФОРТРАН, ПАСКАЛЬ, БЕЙСИК, ФОКАЛ.

Тест-мониторная система (ТМ ОС) обеспечивает контроль работоспособности устройств, входящих в состав ДВК-2М.

Совместимость ДВК по системе команд и системной магистрали с микропроцессорными комплектами серий 581, 588, 1801 позволяет использовать его в качестве инструментальной ЭВМ при разработке и отладке программного обеспечения перечисленных МП. Для осуществления в этом случае программно-аппаратурной отладки на пользователя ложится задача сопряжения системной магистрали и отлаживаемого МПУ. Функциональные возможности по отладке при этом ограничены в первую очередь за счет того, что эмулируется не собственно МП, а соответствующая ему магистраль.

Для подключения к ДВК-2М дополнительного оборудования необходим кабель с коннектором К5, который устанавливается в блок сопряжения и подключается к МПИ. Кабель с коннектором К5 позволяет вывести наружу из ДВК магистраль МПИ и использовать ее в качестве системной магистраль для подключения дополнительного оборудования УОК (блоки эмуляторов, программатор и т. п.). Максимальная скорость обмена информацией по магистрали МПИ составляет около 400К байт/с при программном обмене и около 800К байт/с при обмене в режиме прямого доступа к памяти.

2.3. Блок эмулятора 8-разрядного микропроцессора КР580ВМ80

Архитектурные особенности микропроцессора КР580ВМ80. Микропроцессор КР580ВМ80 предназначен для построения вычислительных устройств, контроллеров, микроЭВМ и мультимикропроцессорных систем управления и обработки цифровых данных. Фиксированная разрядность 8 бит с возможностью обработки слов, кратных байту, система команд, включающая широкий набор арифметических и логических команд, команд управления и пересылок, узел обработки прерываний и развитый интерфейс определяют основное применение этого МП как центрального процессорного устройства параллельной обработки данных. Архитектура МП КР580ВМ80 однозначно определяет архитектуру устройств построенных на его основе [10].

Основные технические характеристики МП КР580ВМ80:

Разрядность	8
Число команд	78
Адресация	Прямая, регистровая, регистровая косвенная, непосредственная, по указателю
Число регистров общего назначения	6, а также аккумулятор и 16-разрядные указатель стека и счетчик команд
Объем адресуемой памяти, К байт	64
Число подключаемых устройств ввода-вывода	256/256
Система прерывания	Одноуровневая восьмивекторная
Тактовая частота, МГц	2,0

Назначение выводов МП КР580ВМ80, дающее самое общее представление о его структуре, приведено в табл. 2.1.

В архитектуре МП КР580ВМ80 отсутствуют какие-либо средства, специально предназначенные для целей отладки. Тем не менее имеются особенности архитектуры, упрощающие построение внутрисхемного эмулятора для МПУ на его базе.

Информация состояния. Действия, выполняемые МП в течение конкретного машинного цикла, идентифицируются 8-разрядной информацией состояния, которая выдается на шины данных в такте Т1 каждого машинного цикла. Информация состояния (слово состояния МП) используется для формирования сигналов управления в микропроцессорной системе, определяя события, которые должны произойти в течение машинного цикла. В частности, значение бита D5 свидетельствует о том, что в МП принимается код операции, т. е. первый байт команды.

Таблица 2.1

Номер вывода	Обозначение вывода	Назначение вывода
1, 25–27, 29–40	A0–A15	Канал адреса
2	GND	Общий
3–10	D0 – D7	Канал данных
11	V_{BB}	Напряжение питания $u_{\text{п}} = -5$В
12	RESET	Установка
13	HOLD	Захват
14	INT	Запрос на прерывание
15, 22	F2,F1	Неперекрывающиеся фазовые импульсы
16	INTE	Разрешение прерывания
17	DBIN	Прием
18	WR	Выдача
19	SYNC	Синхронизация
20	V_{CC}	Напряжение питания $u_{\text{п}} = 5$В
21	HLDA	Подтверждение захвата
23	READY	Готовность
24	WAIT	Ожидание
28	V_{DD}	Напряжение питания $u_{\text{п}} = 12$В

Распознавание этого машинного цикла, во-первых, упрощает реализацию функции дизассемблирования, так как становится непосредственно известен код операции, подлежащий дальнейшей программной дешифрации.

Во-вторых, упрощается переход на служебную программу, так как подстановка команды перехода должна выполняться именно в цикле ввода кода операции.

Сигнал ГОТОВНОСТЬ. Напряжение логического нуля на входе READY (ГОТОВНОСТЬ) свидетельствует о том, что адресуемая память или внешнее устройство не готово для передачи данных. Сигнал используется для синхронизации обмена информацией МП с медленно-действующими запоминающими и внешними устройствами. Если после выдачи адреса на входе присутствует напряжение логического нуля, то МП переходит в состояние ожидания и остается в этом состоянии, пока на вход не поступит логическая единица. Сигнал может быть использован для организации следующих режимов работы МП: поциклового, пошагового (покомандного), останова по адресу. Сигнал может также управлять остановом МП на время загрузки служебной программы, если последняя записывается в ОЗУ программ блока эмулятора.

Программная доступность внутренних ресурсов МП. К числу аппаратных ресурсов МП, находящихся в распоряжении программиста, относятся: аккумулятор, шесть 8-разрядных регистров общего назначения, указатель стека, счетчик команд и регистр флаж-

ков. Все перечисленные ресурсы доступны для ввода и вывода через системную магистраль МП. Это позволяет просто организовать индикацию и модификацию содержимого рабочих регистров МП с помощью служебной программы.

Основные технические характеристики блока эмулятора. Блок эмулятора БЭ-580 представляет собой физическую модель целевого МП КР580ВМ80 и памяти МПУ на его базе и обладает следующими техническими характеристиками:

Вид эмуляции	Внутрисхемная
Объем эмулируемой памяти, К байт	16–32
В том числе:	
объем фиксированной памяти	12–28
объем перемещаемой памяти	4
Максимальный объем отлаживаемых программ, К байт	64
Максимальное число одновременно задаваемых точек останова по адресу	32 К
Разрядность памяти программ и контрольных точек, бит	9
Отладочные функции:	
возможность отладки ПС и АС в реальном масштабе времени	Имеется
условия останова (прерывания эмуляции)	По контрольной точке (по адресу), после выполнения очередной команды или машинного цикла (пошаговый или поцикловый режим), по команде оператора
режим трассировки	Имеется
возможность индикации и модификации содержимого памяти эмулятора или МПУ и внутренних регистров МП	Имеется
возможность оперативного изменения конфигурации системы	Имеется
Типы интерфейсов связи:	
с БВК	Межмодульный параллельный интерфейс МПИ
связь с МПУ	В соответствии со спецификацией на выводы МП
Частота тактового генератора, МГц:	
внутреннего	2,06
внешнего	0,5–2,06
Потребляемая мощность при номинальных напряжениях источников питания, Вт, не более	20
Габаритные размеры, мм, не более	270×320×55

Особенностями блока эмулятора БЭ-580 являются сравнительно большой объем эмулируемой памяти (до 32К байт) и большое число контрольных точек (адресов останова). Для хранения последних в БЭ-580 предусмотрен специальный девятый разряд памяти эмулятора, поэтому максимальное число одновременно хранимых адресов останова равно числу адресуемых ячеек (32К) памяти. Это бывает удобно при отладке сложных и разветвленных программ, так как большое число хранимых адресов контрольных точек (КТ) позволяет пометить все возможные входы и выходы в (из) отлаживаемой программы (подпрограммы).

Конструктивно блок эмулятора БЭ-580 выполнен в виде двух печатных плат: устройства управления и устройства памяти. Плата памяти обеспечивает хранение информации емкостью 16–32 К байт в зависимости от установленного на ней числа БИС ОЗУ.

Положение перемещаемого банка памяти объемом 4К байт в адресном пространстве МП задается оператором программно-аппаратурным способом с дискретностью 4 К.

Возможность оперативного изменения конфигурации системы включает: работу МП с памятью эмулятора или МПУ, работу МП с тактовым генератором эмулятора (внутренним) или тактовым генератором МПУ (внешним), задание местоположения перемещаемого банка ОЗУ блока эмулятора в адресном пространстве МП. Работа с памятью МПУ возможна только при условии использования внешнего тактового генератора.

Структура блока эмулятора. Схема блока эмулятора БЭ-580 приведена на рис. 2.4.

В состав БЭ-580 входят следующие основные функциональные устройства: блок приема-передачи; дешифратор адреса (ДША); дешифратор управляющих сигналов (ДШУС); ОЗУ программ пользователя (ОЗУП) и ОЗУ контрольных точек (ОЗУ КТ); регистр адреса перемещаемого ОЗУП; блок мультиплексоров; буфер данных (БДО); регистр адреса ОЗУП; буфер адреса (БАО); МП КР580ВМ80; регистр слова состояния МП; шинный контроллер; буфера адреса *БА1* и данных *БД1;* блок коммутации управляющих сигналов; блок пуска-останова; тактовый генератор (ТГ).

Блок приема-передачи, ДША и ДШУС образуют интерфейс связи с шиной МПИ БВК, служащий для обмена информацией с БВК и формирования необходимых сигналов управления всем остальным оборудованием БЭ-580 в зависимости от получаемых через шину МПИ командных слов. На рис. 2.4 сигналы управления показаны короткими горизонтальными стрелками без обозначений. Обмен информацией (адреса, данные, команды управления) между БЭ-580 и БВК производится через блок приема-передачи, подключенный к шине МПИ КДА0–КДА15.

Следует иметь в виду, что шина КДА0–КДА15 является двунаправленной и совмещает передачу как адресов, так и данных. При обмене информацией с БВК по шине МПИ КДА0–КДА15 сначала выдается 16-разрядный адрес, а затем вводимые или выводимые данные. Более подроб-

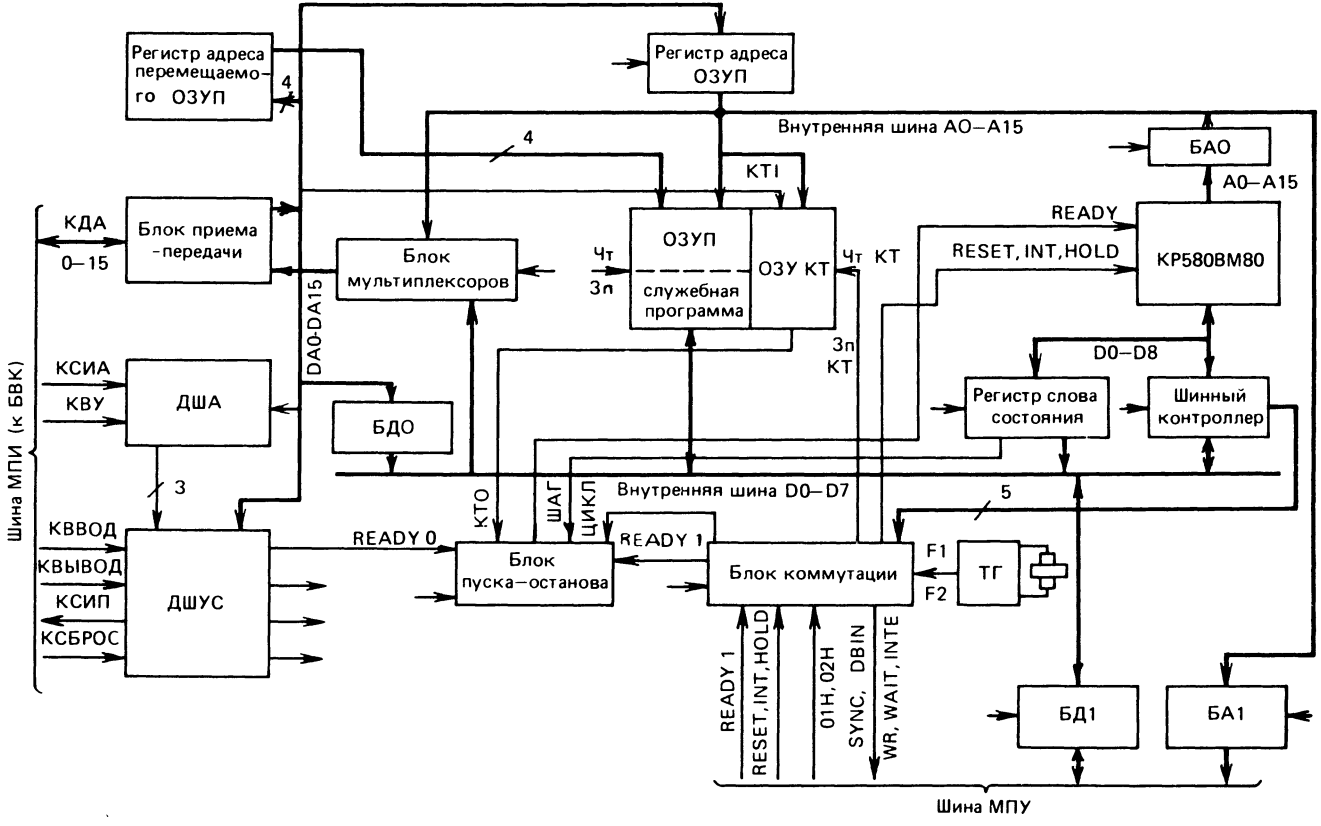


Рис. 2.4. Структурная схема блока эмулятора БЭ-580

но временны́е диаграммы сигналов, формируемых на магистрали МПИ в различных режимах работы, описаны в [7].

Микропроцессор КР580ВМ80, ОЗУП, буфер адреса БАО, шинный контроллер, регистр слова состояния МП и тактовый генератор образуют целевую микроЭВМ, являющуюся основой БЭ-580 и выполняющую функции эмуляции МП и памяти пользователя. Оперативное ЗУП разделено на две части: 4К байт памяти, перемещаемой по адресному пространству (64К байт), и от 12К до 28К байт фиксированной памяти с нулевым начальным адресом.

Регистр адреса ОЗУП и ОЗУ КТ, регистр адреса перемещаемого ОЗУП, блок мультиплексоров и БДО образуют устройство сопряжения целевой микроЭВМ с интерфейсом связи с МПИ.

Блок пуска-останова и ОЗУ КТ служат для управления прерыванием эмуляции в различных режимах (пошаговом, непрерывном режиме с остановом по контрольной точке, поцикловом). Управление режимами работы МП производится сигналами управления, выдаваемыми ДШУС.

Отметим, что ОЗУ КТ представляет собой одноразрядную память, имеющую общий адресный регистр с ОЗУП. Признак останова МП по заданному адресу хранится в этой ячейке ОЗУ КТ в виде логической единицы, отсутствие останова – в виде логического нуля.

Буфера *БА1* и *БД1* и блок коммутации управляющих сигналов образуют интерфейс связи целевой микроЭВМ с шиной отлаживаемого МПУ. Шина связи БЭ-580 с МПУ представляет собой буферированную шину МП КР580ВМ80А. Состав и назначение сигналов этой шины представлены в табл. 2.1. Эмулятор БЭ-580 соединяется с МПУ при помощи кабеля со специальной вилкой, устанавливаемой в панельку МП на плате МПУ.

Связь различных функциональных устройств БЭ-580 осуществляется по внутренним шинам адреса и данных. Доступ к адресной шине БЭ-580 возможен как со стороны БВК (через блок приема-передачи и регистр адреса ОЗУП), так и со стороны целевого МП (через буфер адреса БАО).

Доступ к шине данных БЭ-580 возможен от двух источников: либо со стороны БВК (через блок приема-передачи и буфер данных БДО при записи данных или через блок приема-передачи и блок мультиплексоров при считывании данных), либо со стороны целевого МП (через шинный контроллер). Шина адреса является однонаправленной, а шина данных – двунаправленной.

Естественно, что доступ к шинам адреса и данных производится от этих двух источников поочередно: сначала от БВК, затем от МП или наоборот. Со стороны БВК доступ к шинам адреса и данных БЭ-580 используется для загрузки в ОЗУП программы пользователя или адресов контрольных точек в ОЗУ КТ либо для считывания в БВК содержимого ОЗУП и ОЗУ КТ.

Со стороны МП доступ к шинам адреса и данных используется при выполнении программы пользователя, хранящейся в ОЗУП, в пошаговом, поцикловом или непрерывном режиме работы.

Для контроля за состоянием шин адреса и данных БЭ-580 используется блок мультиплексоров, представляющий собой фактически коммутатор двух входных шин (адрес, данные) в одну выходную шину, связанную со входами блока приема-передачи.

Содержимое регистра слова состояния МП через внутреннюю шину данных и блок мультиплексоров также может быть передано в блок приема-передачи. Аналогично считывается в БВК и разряд ОЗУ КТ (на структурной схеме не показано). Блоком мультиплексоров управляют сигналы, поступающие от ДШУС.

Управление работой МП (пуск-останов) производится с помощью формирования управляющего сигнала ГОТОВНОСТЬ на входе МП. При подаче на этот вход логического нуля выполнение программы приостанавливается, и МП переводится в цикл ожидания, из которого он выводится при установке сигнала ГОТОВНОСТЬ (логическая единица).

Формирование низкого уровня сигнала ГОТОВНОСТЬ выполняется блоком пуска-останова при поступлении на его входы следующих сигналов:

а) выходной сигнал КТО, считываемый из ОЗУ КТ. Сигнал КТО уровня логической единицы является признаком контрольной точки, записанной в ОЗУ КТ по необходимому адресу останова. Следует отметить, что считывание из ОЗУ КТ производится на один машинный такт раньше, чем считывание из ОЗУП, что необходимо для своевременного останова МП;

б) сигнал READY 0, поступающий с выхода ДШУС. Этот сигнал формируется ДШУС в случае необходимости останова по команде, выдаваемой оператором;

в) сигнал READY 1, поступающий через блок коммутации из МПУ. Этот сигнал может использоваться для управления работой МП со стороны МПУ;

г) сигнал ШАГ, поступающий с выхода D5 регистра слова состояния МП. Данный сигнал выдается в начальном машинном цикле M1 при выполнении МП любой команды и используется для пошагового режима работы эмулятора;

д) сигнал ЦИКЛ, формируемый из сигнала SYNC, выдаваемого МП в начале каждого цикла его работы, и тактового сигнала F1. Этот сигнал используется при поцикловом режиме работы.

Управление режимом работы блока пуска-останова производится сигналами, поступающими с выхода ДШУС.

Блок коммутации выполняет следующие функции: подключает к МП внутренний (сигналы F1 и F2) или внешний (сигналы 01Н и 02Н) тактовый генератор; получает от МП выходные сигналы управления (синхронизация, прием, выдача, ожидание, разрешение прерывания), усиливает мощность их и выдает на шину МПУ; принимает от МПУ входные сигналы управления МП (установка, готовность, запрос прерывания, захват) и выдает их на МП или блок пуска-останова.

Шинный контроллер обеспечивает связь шины данных МП с шиной данных БЭ-580, а также формирует сигналы чтения и записи в ОЗУП при обращении к памяти со стороны МП. Сигналы чтения-записи из памяти могут быть сформированы также ДШУС при обращении к памяти со стороны БВК. Переключение сигналов чтения-записи из памяти, поступающих из этих двух источников, производится блоком коммутации.

Кроме того, шинный контроллер формирует управляющие сигналы обращения к портам и устройствам выдачи команд прерывания, расположенным на отлаживаемом МПУ. Эти сигналы преобразуются блоком коммутации в управляющие сигналы DBIN и WR, посылаемые на МПУ.

При работе эмулятора в режиме использования памяти МПУ блок коммутации создает пассивные уровни управляющих сигналов на ОЗУП и преобразует управляющие сигналы, поступающие непосредственно от МП в сигналы DBIN и WR, подаваемые на МПУ.

Регистр адреса перемещаемого ОЗУП управляет четырьмя старшими разрядами полного адреса ОЗУП, что позволяет перемещать физический блок памяти объемом 4К байт по адресному пространству 64К байт с дискретностью, равной 4К. Управление и загрузка регистра адреса перемещаемого ОЗУП осуществляются от БВК через блок приема-передачи. Наличие перемещаемого блока ОЗУП позволяет упростить поиск неисправностей в ОЗУ МПУ благодаря возможности подмены блока памяти МПУ аналогичным (заведомо исправным) блоком памяти, имеющимся в БЭ-580.

С целью лучшего уяснения взаимодействия узлов блока эмулятора в качестве примера рассмотрим выполнение загрузки отлаживаемой программы из БВК в ОЗУП и адресов контрольных точек из БВК в ОЗУ КТ.

Описание процедуры загрузки отлаживаемой программы:

1) остановить выполнение программы МП, для чего с помощью ДШУС выдать сигнал READY 0, который в свою очередь приведет к выдаче блоком пуска-останова неактивного значения сигнала READY на МП;

2) отключить БАО от шины адреса, а также шинный контроллер и регистр слова состояния МП от шины данных, для чего с помощью ДШУС перевести их выходы в состояние с высоким выходным сопротивлением;

3) загрузить из БВК в регистр адреса ОЗУП начальный адрес ОЗУП, начиная с которого производится загрузка отлаживаемой программы;

4) загрузить в регистр адреса перемещаемого ОЗУП четыре старших разряда полного адреса ОЗУП, определяющие местоположение перемещаемого ОЗУП в адресном пространстве 64К байт;

5) подключить буфер данных БДО к внутренней шине данных БЭ-580 и через блок приема-передачи загрузить в БДО байт данных, записываемых по начальному адресу в ОЗУП;

6) выдать с помощью ДШУС команду Зп – запись данных по начальному адресу в ОЗУП;

7) загрузить из БВК в регистр адреса адрес следующей ячейки памяти ОЗУП;

8) продолжить выполнение п. 5–7 до тех пор, пока не будет загружена вся программа;

9) подключить БАО к шине адреса, шинный контроллер и регистр слова состояния – к шине данных, предварительно отключив от них регистр адреса ОЗУП и БДО, а затем снять сигнал READY 0.

Микропроцессор начнет выполнение загруженной программы с текущего адреса, сохранившегося в его счетчике команд. Алгоритм загрузки ОЗУ КТ в основном аналогичен алгоритму загрузки ОЗУП, за исключением следующего.

1. Буфер данных БДО не используется для загрузки контрольных точек, так как имеются отдельные входные (КТ1) и выходные (КТ0) линии данных.

2. Для записи адресов контрольных точек ДШУС выдает команду Зп КТ, в ОЗУП команда Зп не выдается.

Алгоритм функционирования блока эмулятора. Вывод содержимого внутренних регистров и их модификация производятся МП при выполнении служебной программы. Для сокращения аппаратурных затрат постоянное хранение этой программы осуществляется в ОЗУ БВК. После останова МП в пошаговом, поцикловом режимах или по КТ служебная программа вводится из БВК в ОЗУП блока эмулятора. Для хранения служебных программ в ОЗУП отведена специальная область памяти. Чтобы снять ограничения, налагаемые при этом на использование ОЗУП пользователем, содержимое этой части памяти перед засылкой служебной программы предварительно считывается и запоминается в БВК, а после выполнения служебной программы содержимое ОЗУП восстанавливается (вновь загружается из БВК).

Естественно, что пользователь не замечает операции по запоминанию и восстановлению содержимого ОЗУП и может использовать всю имеющуюся в блоке эмулятора память без каких-либо ограничений.

Переход на выполнение служебной программы производится по команде безусловного перехода, которая записывается в три последовательные ячейки ОЗУП, начиная с точки останова. Данные из этих трех ячеек предварительно запоминаются в БВК, а затем восстанавливаются указанным выше способом.

Выполняется служебная программа в пошаговом режиме, что необходимо для ее останова в нужной точке после вывода содержимого внутренних регистров МП в ОЗУП. Базовый вычислительный комплекс считывает содержимое внутренних регистров из ОЗУП, индицирует его на экране дисплея и, если от оператора поступила команда на модификацию внутренних регистров, записывает их новое значение в соответствующие ячейки ОЗУП. Затем БВК производит необходимое число запусков МП в пошаговом режиме, в результате чего МП, продолжая выполнение служебной программы, восстанавливает из ОЗУП содержимое внутренних регистров и значение счетчика команд.

Очевидно, что вывод содержимого внутренних регистров МП возможен только в том случае, если останов МП при выполнении отлаживаемой программы произошел в момент выборки первого байта команды. В противном случае подстановка команды безусловного перехода на служебную программу не имеет смысла и вывод содержимого регистров общего назначения невозможен, о чем дается соответствующее сообщение на экране дисплея БВК.

Чтение содержимого памяти МПУ осуществляется в два этапа. На первом этапе МП блока эмулятора выполняет соответствующую служебную программу по пересылке содержимого памяти МПУ в ОЗУП в поцикловом режиме. При этом в цикле чтения команды МП подключается к ОЗУП, в цикле чтения данных – к памяти МПУ, а в цикле записи данных – к ОЗУП. На втором этапе БВК считывает необходимые данные из ОЗУП и индицирует их на экране дисплея. Аналогично осуществляется запись информации в ОЗУ МПУ.

Блок эмулятора обеспечивает возможность проверки АС МПУ в статическом режиме. Этот режим отладки обеспечивается тем, что после останова МП его выходные сигналы „защелкиваются" в выходных буферах интерфейса связи с МПУ и сохраняются постоянными на все время выполнения служебных операций. Статический режим весьма удобен при поиске неисправностей в МПУ в части системной магистрали, памяти, портов ввода-вывода и логики управления.

Блок эмулятора обеспечивает выполнение функций дизассемблирования. Выполнение этой функции реализуется путем считывания в БВК кода команды с внутренней шины данных МП в момент его останова, дешифрации кода и индикации соответствующей мнемоники на экране дисплея. Функция дизассемблирования выполняется только в пошаговом и поцикловом режимах.

Блок эмулятора обеспечивает защиту входов МПУ от подачи электрических сигналов со стороны БЭ при отсутствии напряжения питания на МПУ. Защита осуществляется переводом выходных буферов блока эмулятора в третье состояние. В качестве сигнала управления используется питание +5В интерфейса связи с МПУ.

Совместное функционирование БВК и блока эмулятора поддерживается управляющей программой DEB580, которая выполняется базовым вычислительным комплексом ДВК-2М в среде операционной системы ОС ДВК. В соответствии с программой DEB580 базовый вычислительный комплекс опрашивает клавиатуру, анализирует команды оператора, формирует управляющие команды и пересылает их, а также адреса и данные в блок эмулятора и, наконец, вводит из блока эмулятора информацию о состоянии его узлов и индицирует ее на экране дисплея.

Схема управляющей программы DEB580 представлена на рис. 2.5. В начале работы программы DEB580 осуществляется очистка экрана дисплея БВК и вывод сообщения *Схемный эмулятор КР580ВМ80.*

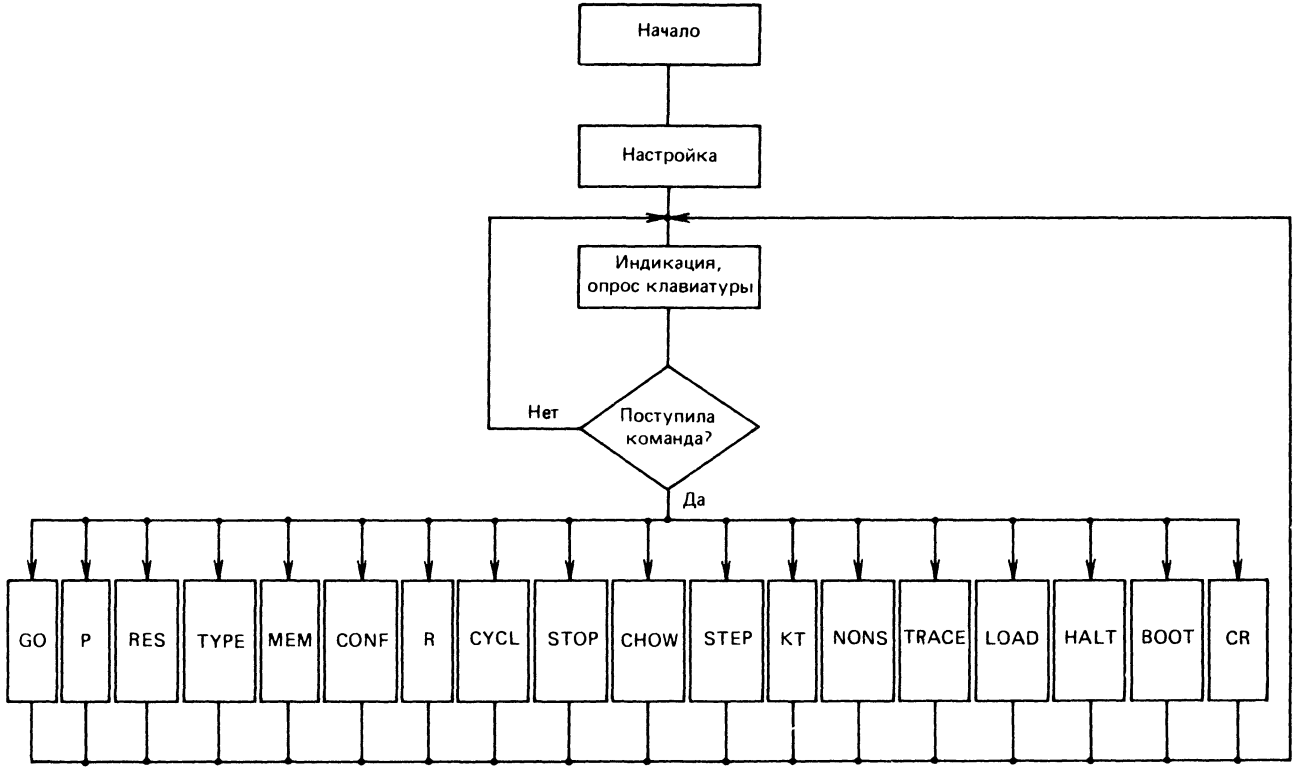


Рис. 2.5. Схема программы DEB580

На следующем этапе выполняется начальная настройка АС блока эмулятора. Записью соответствующего управляющего слова в дешифратор управляющих сигналов по адресу регистра команд назначаются память блока эмулятора, внутренний тактовый генератор, поцикловый режим работы, запрещается общение с МПУ. В ОЗУП блока эмулятора загружается служебная программа, и МП запускается на выполнение требуемого числа циклов служебной программы с целью его установки в начальное состояние – регистры общего назначения и флаги условий обнуляются.

С помощью пробной записи и последующей проверки определяется объем памяти блока эмулятора. Соответствующее сообщение выдается на дисплей.

Завершается настройка диалогом с оператором, в ходе которого определяется конфигурация АС блока эмулятора: задается адресное пространство перемещаемого банка памяти, назначаются тактовый генератор, память и режим работы МП. Далее управляющая программа индицирует на экране символ *, означающий приглашение оператору ввести команду, и циклически опрашивает клавиатуру в ожидании команды.

Поступившая команда дешифрируется и обрабатывается соответствующей подпрограммой обслуживания. Каждая подпрограмма обслуживания представляет собой последовательность обращений БВК по адресам блока эмулятора с целью либо считывания данных или информации о состоянии блока эмулятора, либо записи данных или управляющих слов. По окончании подпрограммы обслуживания управляющая программа индицирует сообщение о результатах выполнения команды и вновь переходит на опрос клавиатуры. Следует отметить особенность обслуживания команд на запуск микропроцессора – GO, P и TRACE. Перед переходом на опрос клавиатуры управляющая программа в этих случаях в течение некоторого времени (около 0,5 с) анализирует состояние сигнала МП ОЖИДАНИЕ. Если сигнал не появляется, то делается вывод о том, что условие останова не выполнялось, и МП выполняет программу в непрерывном режиме.

Полный перечень команд оператора и соответствующие им функции и индикация подробно рассмотрены в §3.1. Полный текст управляющей программы DEB580 приведен в приложении. При работе с текстом управляющей программы следует иметь в виду, что блок эмулятора представляется со стороны БВК как четыре ячейки памяти: регистр адреса (165000_8), регистр данных (165002_8), регистр состояния блока эмулятора (165004_8) и регистр команд (165006_8). Формат слова состояния, считываемого по адресу 165004, представлен в табл. 2.2, а формат управляющего слова, записываемого по адресу 165006, приведен в табл. 2.3. Отметим, что разряд 11 слова состояния и управляющего слова программой DEB580 не анализируется и зарезервирован для целей расширения состава аппаратурных средств блока эмулятора. Методика работы оператора с программой DEB580 изложена в §3.1.

Таблица 2.2

Разряд	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Активный уровень	0	1	1	1	0	1	1	1	1 0	1 0	1 0	1 0	1 0	1 0	1 0	1 0
Информация	Подключена память эмулятора	Подключен внутренний генератор	Организован поцикловый режим	Организован пошаговый режим	Разрешение использования анализатора	Сигнал ГОТОВНОСТЬ МПУ	Сигнал ОЖИДАНИЕ	Требование освобождения шин	D7	D6	D5	D4	D3	D2	D1	D0

Таблица 2.3

Разряд	15		14		13		12		11		10		9	8		2	1	0
Активное значение	1	0	1	0	1	0	1	0	1	0	1	0	X	1	0	0	0	0
Информация	Запрос памяти пользователя	Запрос памяти БЭ-580	Запрос внутреннего генератора	Запрос внешнего генератора	Запрос поциклового режима	Отмена поциклового режима	Запрос пошагового режима	Отмена пошагового режима	Запрет использования анализатора	Разрешение использования анализатора	Запрет общения БЭ-580 с МПУ	Разрешение общения БЭ-580 с МПУ		Требование освобождения шин	Возврат шин микропроцессору			

Примечание. X – безразличное состояние.

В заключение отметим, что моделирование МП КР580ВМ80 блоком эмулятора осуществляется на функциональном уровне. При этом на пользователя не налагается практически никаких ограничений по использованию программных или аппаратурных ресурсов МП. Однако блок эмулятора имеет ряд особенностей, которые надо иметь в виду в процессе отладки МПУ.

Электрические характеристики МП воспроизводятся блоком эмулятора БЭ-580 с некоторыми отклонениями. Применение выходных буферов приводит к тому, что нагрузочная способность выходов эмулятора в 7—10 раз выше, а минимальное значение высокого уровня напряжения для выходных и входных управляющих сигналов на 1,3 В ниже, чем у МП.

Временные характеристики МП КР580ВМ80А воспроизводятся его физической моделью также с некоторыми отклонениями, состоящими в том, что выработка выходных сигналов происходит с задержкой относительно нормального расположения фронтов. Величина этой задержки не превышает 100 нс. Для обеспечения возможности использования отладочного комплекса схема МПУ должна допускать указанную величину сдвига.

2.4. Блок эмулятора 8-разрядной однокристальной микроЭВМ КМ1816ВЕ48

Архитектурные особенности однокристальной микроЭВМ КМ1816ВЕ48. Интегральная схема с большой степенью интеграции КМ1816ВЕ48 представляет собой ОЭВМ (в которой на одном кристалле размещены 8-разрядный ЦП, ППЗУ с ультрафиолетовым стиранием для хранения программ, ОЗУ данных), развитую систему ввода-вывода, включающую 27 линий, 8-разрядный таймер-счетчик событий, тактовый генератор, блок синхронизации, схемы обработки прерываний и начальной установки [5].

Развитая система команд (включающая большой набор команд переходов, команды проверки содержимого отдельных битов аккумулятора или линий ввода и команды установки или сброса отдельных линий ввода-вывода), простая векторная система прерываний и невысокая стоимость позволяют использовать ОЭВМ в качестве контроллера в устройствах самого различного назначения.

Основные технические характеристики БИС КМ1816ВЕ48:

Разрядность	8
Число команд	96
Адресация	Прямая и косвенная (регистровая)
Объем:	
внутренней памяти программ (ПЗУ), К байт	1
внешней адресуемой памяти программ, К байт	4

внутренней памяти данных, К байт	64
внешней памяти данных, байт	256
Число линий ввода-вывода	27, из них 24 сгруппированы в три 8-разрядных порта ввода-вывода
Система прерываний	Два векторных маскируемых прерывания
Тактовая частота, МГц	1–6
Потребляемая мощность, мВт	600

Назначение выводов ОЭВМ типа КМ1816ВЕ48 представлено в табл. 2.4.

Рассмотрим архитектурные особенности ОЭВМ КМ1816ВЕ48, наиболее тесно связанные с разработкой блока эмулятора.

П о р т ы О Э В М. В архитектуре ОЭВМ имеются три 8-разрядных порта: P0 – двунаправленный; P1 и P2 – квазидвунаправленные. Порт P0 представляет собой двунаправленный порт с тремя состояниями. Порт P0 может использоваться либо в качестве выходного канала со статическим буферным регистром, либо в качестве входного канала без буферизации. Информация, выдаваемая портом P0, сопровождается

Таблица 2.4

Номер вывода	Обозначение вывода	Назначение вывода
1	T0	Тестируемый вход 0
2, 3	CRI, CR2	Входы переключения кварцевого резонатора
4	$\overline{\text{CLR}}$	Сброс
5	$\overline{\text{SS}}$	Пошаговый режим
6	$\overline{\text{INT}}$	Прерывание
7	$\overline{\text{EMA}}$	Выборка из внешней памяти
8	R	Чтение
9	PME	Считывание из внешней памяти
10	W	Запись
11	ALE	Фиксация адреса
12–19	DB0–DB7	Шина данных (Порт P0)
20	GND	Общий
21–24, 35–38	P20 – P27	Порт P2
25	PROG	Строб расширителя ввода-вывода
26	U_{DD}	Напряжение питания при программировании
27–34	P10–P17	Порт P1
39	$\overline{\text{T1}}$	Тестируемый вход 1
40	U_{SS}	Напряжение источника питания

строб-импульсом W, а при считывании информации в порт P0 вырабатывается строб-импульс R. Наличие строб-импульсов, сопровождающих передачу информации через порт P0, а также двунаправленность порта P0 делают достаточно простой задачу его эмуляции.

Порты P1 и P2 имеют идентичные характеристики. Данные, выведенные в эти порты, статически защелкиваются и остаются неизменными, пока не будут изменены. При использовании в качестве входов эти порты не стробируются, т. е. данные должны присутствовать, пока не будут считаны командой ввода. Входы полностью совместимы с ТТЛ, а выходы могут быть нагружены одной стандартной ТТЛ нагрузкой. Под квазидвунаправленностью этих двух портов подразумевается фактическое отсутствие третьего состояния на их выходных линиях. Данные, выведенные в порт, постоянно присутствуют на его линиях. Если на эти же линии поступают данные еще и извне для считывания в ОЭВМ, то происходит смешивание тех и других данных в отличие от действительно двунаправленного порта P0. Так как входной ток выходов ОЭВМ в состоянии низкого уровня существенно больше выходного тока схем ТТЛ в состоянии высокого уровня, то при смешивании будут превалировать сигналы низкого уровня. Поэтому для использования портов P1 и P2 в качестве входов ОЭВМ должна предварительно вывести логическую единицу по соответствующим линиям этих портов. Порты P1 и P2 устанавливаются в состояние логической единицы также после подачи сигнала $\overline{CLR}$. Свойство квазидвунаправленности следует иметь в виду при эмуляции этих двух портов.

Другая особенность портов ОЭВМ связана с возможностью их использования в двух режимах. Режим выбирают с помощью входного сигнала $\overline{EMA}$. Если на выводе $\overline{EMA}$ задан сигнла низкого уровня, то ОЭВМ работает с внутренней памятью и порты используются как описанные выше статически фиксированные. В дальнейшем этот режим условно будем называть режимом однокристальной микроЭВМ. Если на вывод $\overline{EMA}$ подан сигнал высокого уровня, то БИС работает с внешней памятью. Этот режим условно назовем режимом отладочного кристалла. В этом режиме порт P0 и четыре младших разряда порта P2 используются для формирования 12-разрядного адреса памяти, а порт P0 еще и для ввода команд и данных. Таким образом, с помощью ввода $\overline{EMA}$ меняется функциональное назначение двенадцати выводов БИС. При эмуляции ОЭВМ желательно реализовать оба ее режима работы.

Наиболее сложная особенность портов связана с тем, что при построении внутрисхемного эмулятора ОЭВМ, очевидно, используется в нем в режиме работы с внешней памятью. В этом режиме содержимое порта P0 и младших четырех разрядов портов P2 портится во время выдачи адреса внешней памяти. Причем если вывод данных в порт P0 сопровождается строб-импульсом W и содержимое P0 может быть просто сохранено во внешнем буферном регистре, то специальный строб-импульс для сопровождения записи данных в порт P2 в архитектуре ОЭВМ отсутствует.

Это вынуждает принимать специальные меры для того, чтобы в реальном масштабе времени аппаратными средствами выделить интервал времени, в течение которого данные выводятся в порт P2, и запомнить их в буферном регистре, например используя сигнал ALE.

Режим пошагового выполнения программы. Режим задается с помощью специального входа пошагового режима $\overline{SS}$. При подаче сигнала $\overline{SS}$ низким уровнем ОЭВМ завершает выполнение команды и делает останов. Пошаговый режим дает возможность останова ОЭВМ после выполнения команды независимо от того, является она одно- или двуцикловой, и предусмотрен специально для отладки и проверки программ.

Программная доступность внутренних ресурсов ОЭВМ. В распоряжении программиста находятся следующие ресурсы ОЭВМ: счетчик команд PC, аккумулятор A, регистр состояния программы PSW таймер-счетчик T, три порта P0–P2, две линии ввода-вывода T0 и T1. Информация из всех перечисленных источников может быть выведена из ОЭВМ. Значение счетчика команд находится на шинах порта P0 и четырех младших разрядах порта P2 после останова ОЭВМ. Содержимое аккумулятора может быть выведено из ОЭВМ с помощью команды пересылки A в память внешнего ЗУ данных. Содержимое остальных источников может быть выведено в память внешнего ЗУ данных через A.

Счетчик команд. 12-разрядный PC организован таким образом, что его содержимое увеличивается на единицу после выборки каждого байта из памяти программ и может быть изменено командами перехода или при прерывании. Причем старший разряд PC изменяется только программно с помощью команд SEL MB0 или SEL MB1. Особенность этих команд заключается в том, что старший разряд изменяется не в момент выполнения команды, а при исполнении следующей команды перехода. В случае останова ОЭВМ после команд типа SEL MB и последующего выполнения служебной программы по индикации состояния ОЭВМ может произойти несанкционированное изменение старшего разряда PC в момент выполнения команды перехода. Избежать ошибки подобного рода можно, выполнив в процессе служебной программы пробную команду перехода и сравнив значения PC до и после этой команды. На основании сравнения можно сделать вывод о необходимости коррекции PC.

Таймер-счетчик. В режиме таймера осуществляется подсчет внутренних синхроимпульсов. Внутренние синхроимпульсы получаются путем пропускания основных синхронизирующих импульсов ALE через делитель частоты на 32. Во время выполнения служебной программы сигналы ALE вызывают искажение значения таймера. Содержимое таймера может быть программно скорректировано, но состояние делителя для контроля или управления недоступно. Один из возможных путей решения проблемы заключается в том, что необходимо ввести в служебную программу дополнительные команды NOP и довести количество сигна-

лов ALE в служебной программе до числа, кратного 32. Последующая коррекция позволяет восстановить значение таймера, а состояние делителя восстанавливается автоматически.

Основные технические характеристики блока эмулятора. Блок эмулятора БЭ-816 представляет собой физическую модель целевой ОЭВМ КМ1816ВЕ48 и памяти МПУ на ее базе. Основные технические характеристики:

Вид эмуляции	Внутрисхемный
Объем эмулируемой памяти программ, К байт	4
Объем эмулируемой памяти данных, байт	256
Разрядность шины данных, бит	8
Разрядность шины адреса, бит	12
Максимальное количество одновременно задаваемых точек останова по адресу	4К
Отладочные функции:	
возможность отладки АС и ПС в реальном масштабе времени	Имеется
условия прерывания эмуляции	По контрольной точке (по адресу), пошаговый режим, по команде оператора
режим трассировки	Имеется
возможность индикации и модификации содержимого памяти эмулятора или МПУ и внутренних регистров ОЭВМ	Имеется
возможность изменения конфигурации системы	Имеется
Типы интерфейсов связи:	
связь с БВК	Межмодульный параллельный интерфейс МПИ
связь с МПУ	В соответствии со спецификацией на выводы ОЭВМ
максимальная тактовая частота, МГц	6
максимальная потребляемая мощность, Вт	10
габаритные размеры не более, мм	306×264×35

Особенностью данного блока эмулятора является большое число контрольных точек (адресов останова), что дает дополнительные удобства при отладке сильно разветвленных программ. Для хранения адресов контрольных точек в памяти блока эмулятора, так же как и в БЭ-580, предусмотрен специальный девятый разряд, поэтому максимальное число одновременно хранимых адресов останова равно 4К.

Как уже отмечалось ранее, БИС КМ1816ВЕ48 может использоваться в составе МПУ в двух режимах: в режиме однокристальной микроЭВМ и в режиме отладочного кристалла. При этом меняется функциональное назначение 12 выводов микросхемы. Более того, разработана и выпускает-

ся ОЭВМ КМ1816ВЕ35, представляющая собой вариант КМ1816ВЕ48. Интегральная схема с большой степенью интеграции КМ1816ВЕ35 в отличие от КМ1816ВЕ48 не имеет на кристалле памяти программ и работает только с внешней памятью. Блок БЭ-816 эмулирует работу ОЭВМ КМ1816ВЕ48 в обоих режимах. Иными словами, БЭ-816 эмулирует, по существу, две микросхемы: КМ1816ВЕ48 или КМ1816ВЕ35. Перевод блока эмулятора из одного режима работы в другой осуществляется с помощью группы переключателей, установленных на его печатной плате.

Возможность изменения конфигурации системы кроме изменения режима также подразумевает работу с памятью МПУ или памятью блока эмулятора, внешним или внутренним кварцевым резонатором. Конструктивно блок эмулятора выполнен в виде одной печатной платы.

Структура блока эмулятора. Схема блока эмулятора БЭ-816 приведена на рис. 2.6.

В состав блока эмулятора входят следующие основные функциональные устройства: блок приема-передачи; дешифратор адреса (ДША); дешифратор управляющих сигналов (ДШУС); блок управления обменом информации; блок выработки управляющих сигналов; ОЗУ программ (ОЗУП); ОЗУ контрольных точек (ОЗУ КТ); ОЗУ данных (ОЗУД); буфера *1–5*; ОЭВМ К1816ВЕ48; блок управления ОЭВМ.

Блок приема-передачи, ДША, ДШУС, блоки управления обменом информации и выработки управляющих сигналов образуют интерфейс связи с магистралью МПИ БВК, служащий для обмена информацией с БВК и формирования сигналов управления остальным оборудованием блока эмулятора. Сигналы управления, выдаваемые блоком выработки управляющих сигналов, на рис. 2.6 показаны короткими горизонтальными стрелками. В частности, для управления остановом ОЭВМ используется сигнал $\overline{SS}$, выдаваемый блоком выработки управляющих сигналов при получении соответствующей команды от БВК через блок приема-передачи. Блок приема-передачи служит для обмена информацией с БВК через шины КДА0–15, по которым из МПИ поступает сначала 16-разрядный адрес, а затем данные. Обмен данными производится побайтно (по 8 бит). Итак, 16-разрядный адрес устройств поступает на ДША, старшие 13 разрядов анализируются дешифратором адреса на соответствие адресам блока эмулятора, а младшие 3 разряда используются для управления работой блока выработки управляющих сигналов и блока управления обменом информации.

Блок управления обменом информации служит для ввода-вывода 12-разрядного адреса и 8-разрядных данных для ОЗУП, ОЗУ КТ и ОЗУД и по существу представляет собой регистры адреса и данных. Блок выработки управляющих сигналов представляет собой регистры для хранения управляющих слов, программно записываемых из БВК через блок приема-передачи. Данные блоки построены на базе БИС параллельного программируемого интерфейса КР580 ВВ55.

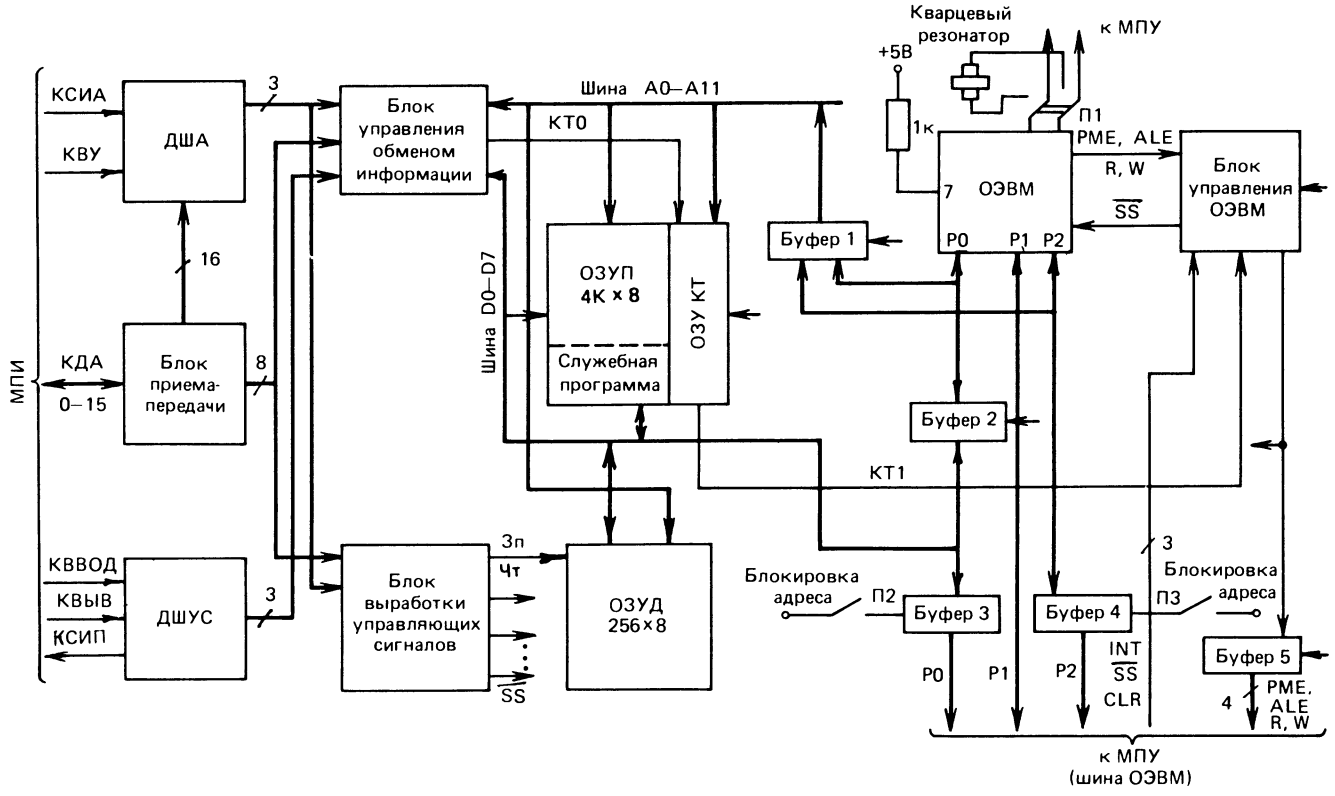


Рис. 2.6. Структурная схема блока эмулятора БЭ-816

В совокупности ОЭВМ, ОЗУП, ОЗУ КТ, ОЗУД, буфера *1–5* и блок управления ОЭВМ образуют целевую микроЭВМ, являющуюся основой блока эмулятора и выполняющую функции эмуляции двух типов ОЭВМ серии К1816, а также функцию эмуляции памяти программ и данных. Оперативное ЗУП и ОЗУД имеют двойной доступ: со стороны БВК через МПИ, блок приема-передачи и блок управления обменом информации и со стороны ОЭВМ через буфера *1* и *2*. Оперативное ЗУ КТ имеет доступ на запись только со стороны БВК, а на чтение – со стороны БВК и ОЭВМ. Оперативное ЗУП и ОЗУ КТ имеют общий адресный регистр, но раздельные информационные входы и выходы.

Однокристальная ЭВМ К1816ВЕ48 используется в составе блока эмулятора в режиме с внешней памятью программ. Для этого на вывод *7* ОЭВМ постоянно подан уровень напряжения +5 В через резистор 1 кОм. Может быть использована и ОЭВМ КМ1816ВЕ35. Совместно с ОЭВМ может работать как внешний, так и внутренний кварцевый резонатор. С целью экономии объема оборудования переключение кварцевых резонаторов (внутреннего или внешнего) производится вручную с помощью переключателя *П1*.

Сигнал $\overline{SS}$, используемый для останова ОЭВМ, может быть выработан одним из следующих источников: блоком выработки управляющих сигналов по команде БВК, выданной с клавиатуры или программно; при считывании контрольной точки из ОЗУ КТ (сигнал КТ1); из МПУ через шину ОЭВМ.

Коммутацию всех этих сигналов и выдачу сигнала $\overline{SS}$ непосредственно на ОЭВМ осуществляет блок управления ОЭВМ. Кроме того, блок управления ОЭВМ выдает на МПУ (через буфер *5*) или на ОЗУП и ОЗУД сигналы управления PME, ALE, R и W, поступающие на его вход из ОЭВМ.

Переключение режима работы эмулируемой ОЭВМ (однокристальная микроЭВМ или отладочный кристалл) производится также вручную с помощью переключателей П2 и П3, которые позволяют выполнять блокировку выдачи из ОЭВМ в МПУ адреса через порты P0 и P2.

Рассмотрим в качестве примера взаимодействия БВК и блока эмулятора перечень операций, которые необходимо выполнить для записи программы в ОЗУП блока эмулятора:

1. Остановить выполнение программы ОЭВМ (блок выработки управляющих сигналов по команде из БВК выдает сигнал $\overline{SS}$), буфера *1–5* перевести в третье состояние (высокий импеданс на выходе), в котором они не оказывают влияния на работу памяти и МПУ.

2. Загрузить из БВК в блок управления обменом информации адрес (12-разрядный) и код команды (8 бит) и выдать с помощью блока управляющих сигналов сигнал разрешения записи в ОЗУП.

3. Загрузить следующий адрес и код команды и произвести запись в ОЗУП. Действия по п. 3 повторяются до тех пор, пока не будет записана последняя команда программы в ОЗУП.

Загрузка ОЗУ КТ производится аналогично, но признак контрольной точки (логическая 1) подается из блока управления обменом информации по отдельной линии (КТО) и блоком управляющих сигналов формируется отдельный сигнал разрешения записи.

Чтение ОЗУП производится аналогично загрузке, с той лишь разницей, что блок управляющих сигналов формирует сигнал разрешения чтения, а данные из памяти поступают на вход блока управления обменом информации и далее через блок приема-передачи в БВК.

Запись и чтение данных в ОЗУД выполняется аналогично п. 1–3 с той лишь разницей, что блок управления обменом информации формирует не 12-разрядный, а 8-разрядный адрес, а блок управляющих сигналов выдает соответствующие сигналы разрешения записи-чтения на ОЗУД.

После загрузки отлаживаемой программы в ОЗУП и контрольных точек в ОЗУ КТ ОЭВМ может работать в следующих режимах, задаваемых пользователем из БВК: пошаговый, с остановом по контрольным точкам, непрерывный.

Для запуска ОЭВМ на выполнение программы необходимо произвести следующие действия:

1. Открыть буфера *1* и *2*, т. е. подготовить их для выдачи данных и адреса памяти из ОЭВМ и отключить выходы регистров адреса и данных, имеющихся в блоке управления обменом информации, переведя их выходы в третье состояние.

2. Снять сигнал $\overline{SS}$, выдаваемый блоком выработки управляющих сигналов, т. е. запустить выполнение программы с текущего адреса, сформированного ОЭВМ.

3. Выполнить команду, считанную ОЭВМ по текущему адресу, и сформировать адрес следующей команды и т. д.

При считывании контрольной точки из ОЗУ КТ, которая поступает в виде сигнала КТ1 на вход блока управления ОЭВМ, выдается сигнал $\overline{SS}$, заканчивается выполнение очередной команды, ОЭВМ останавливается. После останова ОЭВМ БВК загружает в ОЗУП служебную программу, по которой последовательно выводятся на индикацию содержимое внутренних программно-доступных регистров ОЭВМ (аккумулятор, порты, внутренняя память данных и т. д.). Предварительно в памяти БВК запоминаются текущий адрес останова и содержимое той части памяти ОЗУП, которая занята служебной программой индикации. Кроме того, в БВК запоминается также содержимое аккумулятора ОЭВМ в момент останова, что связано со спецификой ОЭВМ серии К1816: содержимое ее внутренней памяти данных, портов и таймера можно вывести на индикацию в БВК только через аккумулятор. При этом, естественно, первоначальное содержимое аккумулятора в момент останова ОЭВМ теряется. После выполнения служебных программ индикации БВК восстанавливает первоначальное состояние аккумулятора, текущий адрес ОЭВМ в момент останова, а также содержимое памяти ОЗУП, занятой служебной программой. Затем программа, выполняемая ОЭВМ, вновь запускается БВК

(снимается сигнал $\overline{SS}$) и все повторяется после считывания следующей контрольной точки.

При пошаговом режиме останов ОЭВМ производится после выполнения каждой команды, потом производятся те же действия по индикации внутренних ресурсов ОЭВМ, что и при останове по контрольной точке. Выбор режима останова (пошаговый или останов по контрольной точке) производится блоком выработки управляющих сигналов, получающим соответствующие команды из БВК.

В основных чертах алгоритм функционирования БЭ-816 повторяет алгоритм функционирования БЭ-580. Совместная работа БВК и блока эмулятора БЭ-816 поддерживается управляющей программой DEB816. Методика работы оператора с программой DEB816 изложена в §3.2.

Блок эмулятора БЭ-816 имеет ряд особенностей, которые следует учитывать при его эксплуатации. В отличие от реальной ОЭВМ К1816ВЕ48, новое значение информации, записанное в порт Р2 блока эмулятора, появляется на его выводах через 100 нс после переднего фронта следующего импульса сигнала ALE.

В режиме однокристальной микроЭВМ команды „ORL BUS, данные" и "ANL BUS, данные" не выполняются. Это связано с тем, что при выполнении указанных команд вывод данных в порт Р0 не сопровождается строб-импульсом $\overline{W}$.

Выводы T0, T1, CR1, CR2, PROG и P10–P17 ОЭВМ в целях организации работы блока эмулятора не задействованы, и сигналы по этим линиям не буферированы. На указанных выводах интерфейса связи с МПУ сигналы подаются непосредственно от ОЭВМ. Остальные сигналы буферированы и имеют ТТЛ-уровни логических нуля или единицы.

2.5. Блок эмулятора 16-разрядного микропроцессора КМ1810ВМ86

Архитектурные особенности микропроцессора КМ1810ВМ86. Микросхема КМ1810ВМ86 представляет собой высокопроизводительный 16-разрядный микропроцессор, программно и аппаратурно совместимый с микропроцессорным комплектом БИС серии К580, однако обладающий на порядок более высокой производительностью [11]. Высокая эффективность МП КМ1810ВМ86 определяется следующими факторами: высоким быстродействием, мощной и гибкой системой команд, большим объемом адресуемой памяти и разнообразием способов адресации, развитой системой прерываний.

Основные технические характеристики МП:

Система команд	135 базовых команд, безадресная, одноадресная, двухадресная; совместим с системой команд БИС КР580ВМ80 на уровне ассемблера

Типы обрабатываемых данных	Биты, байты, слова (2 байта), двоично-десятичные упакованные и распакованные числа, строки длиной до 64К байт
Число программно-доступных регистров .	14 16-разрядных
Объем адресуемой памяти, М байт	1
Число адресуемых устройств ввода-вывода	64К
Число способов адресации операнда в памяти	24
Число типов прерывания	256
Тактовая частота, МГц	До 5
Максимальное быстродействие для команд типа регистр – регистр, операций/с	До $2{,}5 \cdot 10^6$

Назначение выводов микропроцессора приведено в табл. 2.5.

В архитектуре МП КМ1810ВМ86 выделим те ее особенности, которые в первую очередь связаны с архитектурой аппаратурных средств отладки.

К о н в е й е р н ы й р е ж и м р а б о т ы. Микропроцессор КМ1810ВМ86 реализован по традиционной архитектуре, в соответствии с которой он последовательно получает из внешней памяти команды и данные и выполняет эти команды в соответствии со своими вычислительными возможностями. Структура МП отличается наличием двух основных частей: устройства обработки (УО) и устройства сопряжения с каналом (УСК).

Устройство обработки осуществляет декодирование и выполнение команд и включает в свой состав: 16-разрядное АЛУ, восемь 16-разрядных регистров общего назначения и регистр признаков. Функции управления выполнением команд возложены на микропрограммное устройство управления, которое декодирует команды и вырабатывает необходимые сигналы управления. Устройство сопряжения с каналом предназначено для осуществления связи с внешними устройствами и обеспечивает выполнение функций, связанных с формированием 20-разрядного физического адреса памяти, выборкой команд и операндов из памяти и записью результатов выполнения команд в память, формированием очереди команд.

Оба устройства независимы, работают асинхронно и обеспечивают конвейерный принцип выполнения команд с предварительной выборкой. Распределение функций выборки и выполнение команд между двумя независимыми частями МП приводит к распараллеливанию работы и повышению его производительности. В УСК организуется предварительная очередь команд, состоящая из трех 16-разрядных слов. В момент, когда УО требуется очередная команда, она уже находится в УСК. Устройство сопряжения с каналом осуществляет выборку следующего слова из памяти, как только в очереди освобождается по меньшей мере 2 байта.

Таблица 2.5

Номер вывода	Обозначение вывода	Назначение вывода
1, 20	GND	Общий
2–16, 39	AD0–AD15	Канал адреса/данных
17	NMI	Немаскируемый запрос прерывания
18	INT	Маскируемый запрос прерывания
19	CLC	Тактовый импульс
21	CLR	Установка
22	RDY	Готовность
23	TEST	Проверка
32	$\overline{R}$	Чтение
33	MN/$\overline{MX}$	Минимальный (максимальный) режим
34	$\overline{BHE}$/ST7	Разрешение передачи по старшей половине канала данных D15 – D8/сигнал состояния 7
35–38	A19/ST6, A18/ST5, A17/ST4 A16/ST3	Канал адреса/сигналы состояния
40	U_{CC}	Напряжение источника питания +5 В
	Минимальный режим	
24	$\overline{INTA}$	Подтверждение прерывания
25	$\overline{STB}$	Строб адреса
26	$\overline{DE}$	Разрешение передачи данных
27	OP/$\overline{IP}$	Выдача/прием данных
28	M/$\overline{IO}$	Память/внешнее устройство
29	$\overline{W}$	Запись
30	HLDA	Подтверждение захвата
31	HLD	Захват
	Максимальный режим	
24, 25	$\overline{QS1}$ $\overline{QS0}$	Сигналы состояния очереди команд
26–28	$\overline{ST0}$, $\overline{ST1}$, ST2	Сигналы состояния цикла канала
29	$\overline{LOCK}$	Канал занят
30, 31	$\overline{RQ}$/EI $\overline{RQ}$/EO	Запрос/разрешение доступа к магистрали

Шестибайтная глубина предварительной очереди позволяет УО практически не простаивать в ожидании выборки команды из памяти. Кроме того, предварительная выборка команд позволяет МП не превращаться в монопольного пользователя системным каналом.

Конвейерный принцип выполнения команд существенно осложняет построение аппаратурных средств отладки, так как в подавляющем большинстве случаев в каждый текущий момент времени выполняемая команда не соответствует значениям сигналов на внешних выводах МП. В частности, при проектировании памяти „следа" необходимо учитывать, что при всяком непоследовательном выполнении команд программы (например, безусловные или условные переходы) очередь команд в УСК опустошается и соответствующие им циклы обращения к каналу стано-

вятся в некотором роде лишними для памяти следа. Серьезные проблемы вызывает также реализация аппаратурного прерывания эмуляции по контрольным точкам или условиям. Решение проблемы может быть достигнуто за счет дублирования очереди команд и восстановления исполняемого адреса, соответствующего байту, выбираемому из очереди команд в УСК. Для полной и эффективной эмуляции в этом случае необходимы существенные аппаратурные затраты.

Многофункциональное использование выводов МП. Одной из отличительных особенностей рассматриваемого МП является возможность аппаратурной перестройки внутренней структуры схемы управления и синхронизации и выбора одного из двух режимов работы: минимального или максимального. Настройка МП на тот или иной режим работы обеспечивается путем подключения специального вывода режима MN/MX либо к выводу источника питания, либо к выводу ОБЩИЙ. При этом меняется функциональное назначение восьми выводов МП.

В минимальном режиме МП используется в системах, имеющих простую конфигурацию. В этом режиме МП вырабатывает все необходимые сигналы управления памятью и периферийными устройствами по выводам 24–31.

Максимальный режим МП используется для построения сложных многопроцессорных систем и применения сопроцессоров специального назначения. В этом режиме три сигнала $\overline{ST0}$ – $\overline{ST2}$ определяют состояние цикла канала. Функции управления каналом берет на себя системный контроллер КР1810ВК88, который дешифрирует сигналы состояния и обеспечивает все выходные сигналы управления обменом вместо МП. Остальные пять сигналов используются для организации совместной работы МП в многопроцессорных системах.

Структура прерываний. Микропроцессор КМ1810ВМ86 имеет простую и гибкую векторную структуру прерывания. Микропроцессор может обрабатывать до 256 типов различных прерываний. Каждому типу прерывания присвоен отдельный номер. Вектора прерываний последовательно занимают область памяти с нулевого адреса до $3FF_{16}$. Прерывания могут быть внутренними и внешними по отношению к МП, программными и аппаратурными, маскируемыми и немаскируемыми.

Среди всех прерываний отметим три типа, имеющих непосредственное отношение к построению отладочных средств.

Прерывание типа 1 – пошаговый режим; это внутреннее аппаратурное немаскируемое прерывание. Если установлен флаг пошагового режима в регистре признаков, то МП автоматически вырабатывает прерывание типа 1 после выполнения каждой команды. Данное прерывание позволяет просто и эффективно организовать покомандное выполнение программы с выводом содержимого всех внутренних регистров МП в процессе отладки. Прерывание типа 2 – немаскируемое внешнее прерывание, осуществляемое по выводу NMI. Обладает самым высоким приоритетом

среди аппаратурных прерываний и обычно используется при катастрофических событиях, таких, например, как обнаружение ошибки памяти.

Прерывание типа 3, иначе называемое прерыванием по точкам разрыва, вызывается специальной командой прерывания длиной в один байт INT3. Данное прерывание не маскируется и обычно используется в программных средствах отладки для перехода на выполнение служебной программы для вывода содержимого регистров и т. п.

Программная доступность внутренних ресурсов МП. Для программиста внутренняя структура МП представляется следующими программно-доступными функциональными блоками: регистры общего назначения, сегментные регистры, регистр адреса команд и регистр признаков. Содержимое всех перечисленных ресурсов программно может быть выведено из МП или введено в него после коррекции через мультиплексированный канал AD.

Ожидание и проверка. Микропроцессор КМ1810ВМ86 в максимальном и в минимальном режимах может быть синхронизирован по отношению к внешним событиям с помощью команды WAIT (ожидание для проверки) и входного сигнала $\overline{\text{TEST}}$. При выполнении МП команды WAIT результат операции зависит от значения сигнала $\overline{\text{TEST}}$. Если сигнал $\overline{\text{TEST}}$ имеет высокий уровень, то МП переходит в холостое состояние и периодически, через каждые пять тактов, опрашивает сигнал $\overline{\text{TEST}}$. Как только сигнал $\overline{\text{TEST}}$ принимает значение низкого уровня, МП переходит на выполнение следующей после WAIT команды.

Основные технические характеристики блока эмулятора. Блок эмулятора БЭ-810 представляет собой физическую модель целевого МП КМ1810ВМ86 и памяти МПУ. Основные технические характеристики:

Вид эмуляции	Внутрисхемный
Объем эмулируемой памяти, К байт:	
фиксированная память	8
перемещаемая память	8
Объем отлаживаемых программ, К байт, не менее	15
Максимальное число одновременно задаваемых точек программного прерывания эмуляции	7
Отладочные функции:	
возможность отладки программ в реальном масштабе времени	Имеется
условия программного прерывания эмуляции	По значению регистра команд, пошаговый режим, по значению регистров МП, по изменению содержимого заданной ячейки ОЗУ, по команде оператора

режим трассировки	Имеется
возможность индикации и модификации содержимого памяти эмулятора или МПУ и программно-доступных регистров МП	Имеется
возможность оперативного изменения конфигурации системы	Имеется
Типы интерфейсов связи:	
связь с БВК	Межмодульный параллельный интерфейс МПИ
связь с МПУ	В соответствии со спецификацией на выводы МП
Максимальная тактовая частота, МГц	5
Напряжение питания, В	(5±5) %
Потребляемая мощность, Вт, не более	20

Блок эмулятора БЭ-810 построен на базе МП КМ1810ВМ86.

Объем эмулируемой памяти блока эмулятора может быть расширен до 256К байт. Положение перемещаемой памяти в адресном пространстве задается оператором программно-аппаратурным способом (1К байт эмулируемой памяти резервируется под служебные цели системы).

В режиме трассировки МП блока эмулятора работает в пошаговом режиме, после выполнения каждой команды осуществляется индикация содержимого всех программно-доступных регистров МП, переход к выполнению следующей команды осуществляется автоматически.

Возможность оперативного изменения конфигурации системы включает: эмуляцию МП в минимальном или максимальном режиме, работу МП с памятью эмулятора или МПУ, работу МП с цепями синхронизации блока эмулятора или МПУ.

Структура блока эмулятора. Схема блока эмулятора БЭ-810 приведена на рис. 2.7.

Интерфейсы блока эмулятора БЭ-810. В структуре блока эмулятора можно выделить три интерфейса, связывающие узлы блока эмулятора между собой и с внешними устройствами: интерфейс связи с БВК, интерфейс связи с МПУ, внутренняя шина адреса/данных (ША/ШД).

Интерфейс связи с БВК соответствует спецификации, временным диаграммам и нагрузочной способности стандартного межмодульного параллельного интерфейса МПИ.

Интерфейс связи с МПУ пользователя соответствует спецификации выводов и временным диаграммам целевого МП КМ1810ВМ86.

Внутренняя шина А/Д включает внутренние шину адреса и шину данных, обеспечивающих доступ к ОЗУ эмулятора как со стороны МП, так и со стороны БВК.

Дешифратор адреса и управляющих сигналов (ДШАУС). Осуществляет обмен информацией между блоком эмулятора и БВК, обеспечивает доступ со стороны БВК к ОЗУ эмулятора, а

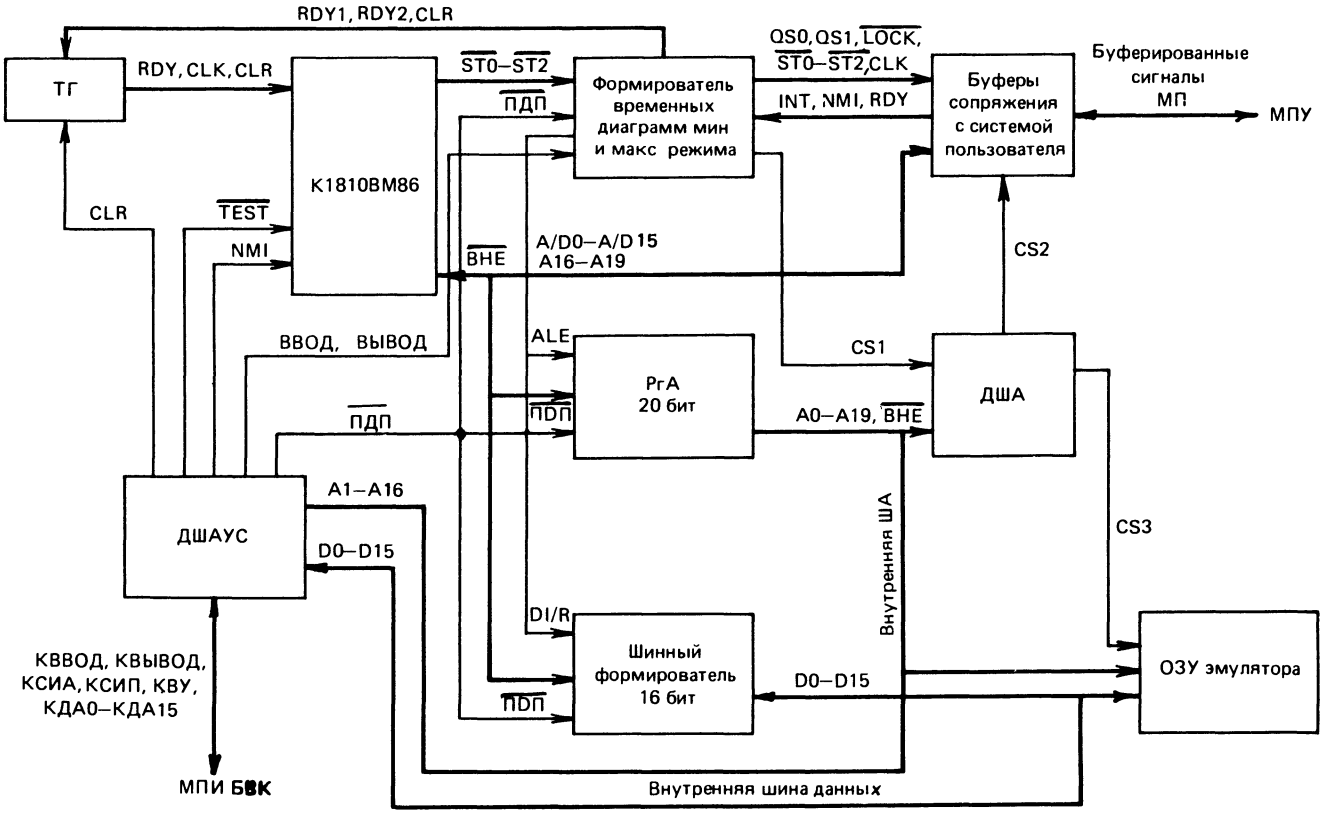


Рис. 2.7. Структурная схема блока эмулятора БЭ-810

также хранение и дешифрацию команд по управлению блоком эмулятора. В основу ДШАУС положена типовая схема, обслуживающая принятый для МПИ протокол обмена в циклах ВВОД и ВЫВОД и включающая шинные формирователи и логику дешифрации адреса, поступающего со стороны МПИ. Дешифратор ДШАУС также содержит набор регистров для хранения команд управления блоком эмулятора, информации о состоянии блока эмулятора и адреса обращения к ОЗУ эмулятора.

Со стороны МПИ блок эмулятора представляется как три адресуемых регистра: команд, адреса и данных. Обращение к ОЗУ эмулятора осуществляется в результате выполнения двух циклов канала МПИ: в первом цикле в регистр адреса РгА заносится адрес ячейки ОЗУ эмулятора, а во втором цикле происходит обмен данными по адресу регистра данных.

Запись в регистр команд соответствующего управляющего слова приводит к выработке одного из следующих сигналов управления: CLR, $\overline{\text{TEST}}$, NMI, $\overline{\text{ПДП}}$, ВВОД, ВЫВОД. Сигнал CLR переводит внутренние узлы процессора в начальное состояние и используется для запуска МП. Сигнал $\overline{\text{TEST}}$ предназначен для вывода МП из состояния холостого хода, в котором он находится после выполнения команды ожидания WAIT в служебной программе (мониторе). Сигнал NMI предназначен для организации аппаратурного прерывания эмуляции. Сигнал $\overline{\text{ПДП}}$ переводит в третье состояние РгА и шинный формирователь и разрешает тем самым доступ ДШАУС к внутренним ША/ШД эмулятора. Сигналы ВВОД и ВЫВОД указывают направление передачи данных по отношению к ДШАУС.

Тактовый генератор (ТГ). Формирует тактовые импульсы частотой 5 МГц, а также синхронизирует сигналы RDY и CLR, необходимые для работы МП и контроллеров шин. Тактовые импульсы и сигнал CLR также подаются в МПУ пользователя. Сигналы RDY поступают на ТГ как со стороны ОЗУ эмулятора, так и со стороны пользователя. Разрешение на работу по тому или иному сигналу RDY вырабатывается ДША по входам активации генератора.

Центральный процессор блока эмулятора. Выполняет команды отлаживаемой и служебной программ, а также формирует сигналы управления блока эмулятора. Центральный процессор используется в блоке эмулятора в максимальном режиме.

Формирователь временных диаграмм минимального и максимального режимов. Выполняется на основе системного контроллера БИС КР1810ВК88. Контроллер декодирует три сигнала состояния МП $\overline{\text{ST0}}$–$\overline{\text{ST2}}$ и формирует необходимый набор сигналов управления обменом, соответствующий минимальному режиму МП. В состав формирователя входит также мультиплексор, подключающий к буферу сопряжения с системой пользователя либо сигналы максимального режима от МП, либо сигналы минимального режима от системного контроллера. Управление мультиплексированием осуществляется сигналом MN/$\overline{\text{MX}}$, поступающим из МПУ.

Буферы сопряжения с системой пользователя В блоке эмулятора БЭ-810 применена архитектура с полной буферизацией, основными достоинствами которой являются минимизация потенциальных повреждений от пользовательской системы и способность комплекса сохранять управление процессом эмуляции и продолжать функционирование даже при наличии неисправности в устройстве пользователя. Управление буфером сопряжения с системой пользователя осуществляется сигналами формирователя временны́х диаграмм, а также дешифратора адреса.

Регистр адреса (РгА). Микропроцессор КМ1810ВМ86 имеет мультиплексированную во времени шину адреса и данных. Регистр адреса позволяет сформировать 20-разрядную внутреннюю ША блока эмулятора. Стробирование адреса в регистре осуществляется формирователем временны́х диаграмм (сигнал ALE). Сигналом $\overline{\text{ПДП}}$ выходы регистра адреса переводятся в третье состояние.

Шинный формирователь. Позволяет реализовать двунаправленную 16 разрядную внутреннюю ШД блока эмулятора. Направление передачи определяется формирователем временны́х диаграмм (сигнал DI/R). Аналогично регистру адреса сигналом $\overline{\text{ПДП}}$ шинный формирователь переводится в третье состояние.

Дешифратор адреса (ДША). С целью сокращения аппаратурных затрат, учитывая большой объем адресуемой микропроцессором памяти, в архитектуре блока эмулятора применен метод разделения адресного пространства между системой пользователя и системой эмулятора. Адресное пространство распределено следующим образом: 0 – 3FFFF – ОЗУ блока эмулятора, 40000 – FFFEF – адреса пользователя, FFFFO – FFFFF – ОЗУ эмулирующей системы. В младших адресах ОЗУ блока эмулятора размещаются отлаживаемые программы, три задействованных в системе вектора прерываний и служебная программа (монитор). В старших адресах ОЗУ блока эмулятора размещается команда безусловного перехода на служебную программу. Практически данный метод не налагает на пользователя серьезных ограничений, поскольку объем отлаживаемых программ для встраиваемых МПУ относительно невелик и редко превышает 16К байт. В соответствии с принятым распределением адресного пространства ДША вырабатывает сигналы выборки ОЗУ эмулятора или памяти МПУ. Соответственным образом происходит выборка сигнала RDY от системы пользователя или ОЗУ эмулятора.

Оперативное ЗУ эмулятора. Схемотехнических отличий от стандартных систем не имеет. Архитектурной особенностью ОЗУ эмулятора является возможность перемещения по крайней мере 8К байт памяти в адресном пространстве ОЗУ. Интервал перемещения составляет 128 байт. Начальный адрес перемещаемого массива хранится в специальном регистре ДША, а заносится туда по команде управляющей программы БВК.

Алгоритм функционирования блока эмулятора. Центральное место при рассмотрении алгоритма функционирования блока эмулятора БЭ-810 занимают прерывание эмуляции, переход на выполнение служебной программы (иначе – монитора) и возврат из нее. Основное назначение монитора заключается в выводе и вводе значений внутренних регистров МП. Кроме того, по командам монитора вырабатывается ряд сигналов, обеспечивающих синхронизацию работы блока эмулятора и БВК.

Прерывание эмуляции и обращение к монитору в блоке эмулятора БЭ-810 осуществляется при выполнении одного из трех типов немаскируемых прерываний: прерывание типа 1 – пошаговый режим, прерывание типа 2 – немаскируемое аппаратное прерывание (NMI), однобайтное программное прерывание типа 3. Значения векторов прерываний перечисленных типов предварительно заносятся в ОЗУ эмулятора с помощью управляющей программы БВК. Переход на выполнение монитора происходит также при начальном пуске блока эмулятора, когда по сигналу CLR МП выбирает первую команду из ячейки памяти с адресом FFFF0H. В эту ячейку ОЗУ эмулятора управляющая программа БВК предварительно заносит команду прямого межсегментного перехода JMP на начало монитора.

Обслуживание прерываний начинается с записи в стек значений регистра признаков, сегментного регистра команд CS и регистра адреса команд IP. Разряд пошагового режима регистра признаков при обработке прерывания устанавливается в неактивное состояние. Новые значения CS и IP определяются соответствующим вектором прерывания.

Выполняя программу монитора, МП выводит содержимое внутренних регистров в служебную область ОЗУ эмулятора. Завершив вывод, МП выполняет команду по установке в регистре состояний ДШАУС разряда, индицирующего переход процессора в состояние ожидания. Обращение к регистру состояний осуществляется так же, как к ячейке памяти с адресом из служебной области адресов. Затем МП выполняет команду WAIT и переходит в холостое состояние.

Базовый вычислительный комплекс циклически анализирует значение регистра состояния. Определив состояние ожидания МП эмулятора, БВК записывает в регистр команд ДШАУС команду на выработку сигнала $\overline{\text{ПДП}}$, считывает из ОЗУ эмулятора и выводит на индикацию значения внутренних регистров МП. Имея доступ к памяти, БВК по команде оператора может вывести на индикацию или модифицировать содержимое любой ее области. Изменяя содержимое служебной области ОЗУ эмулятора, БВК выполняет команды оператора по модификации регистров общего назначения, по изменению конфигурации системы или режима работы. Так, например, для отмены или назначения пошагового режима в стеке в регистре признаков необходимо сбросить или установить разряд пошагового режима. Команда оператора на пуск отлаживаемой программы с заданного адреса выполняется путем занесения указанного адреса в стек и т. д.

Завершая выполнение инструкций оператора, БВК записывает в регистр команд ДШАУС команду на выработку сигнала $\overline{TEST}$. Микропроцессор эмулятора переходит на выполнение следующих после WAIT команд монитора и загружает из ОЗУ эмулятора новое значение внутренних регистров Завершается программа монитора командой возврата из прерывания IRET, по которой значения регистра признаков МП, регистров IP и CS загружаются из стека.

Объем монитора составляет около 0,5К байт. По команде оператора в ходе начального диалога монитор размещается в свободной области ОЗУ эмулятора. Настройка монитора для размещения его в выбранной оператором области памяти выполняется автоматически.

Алгоритм выполнения других команд оператора по управлению работой блока эмулятора очевиден из рассмотрения структурной схемы и не требует дополнительных пояснений.

Рассмотренный вариант технического решения по реализации блока эмулятора налагает на пользователя ряд ограничений, так как часть ресурсов МП использована для организации работы системы с целью сокращения аппаратурных затрат. Ячейки памяти FFFF0H, а также вектора прерываний типов 1–3 задействованы для выполнения переходов на служебную программу. Для размещения монитора и на другие служебные цели используется около 0,5К байт адресного пространства. Внешнее аппаратурное немаскируемое прерывание NMI также задействовано под служебные цели эмулятора.

2.6. Блок эмулятора 8-разрядного микропроцессора К1821ВМ85А

Архитектурные особенности микропроцессора К1821ВМ85А. Микропроцессор К1821ВМ85А представляет собой улучшенный вариант МП КР580ВМ80, обладает аналогичной системой команд и структурой, но содержит ряд аппаратурных усовершенствований, упрощающих его применение в МПУ [12]. Число источников питания сокращено с трех до одного. Прибор К1821ВМ85А имеет встроенный тактовый генератор ТГ, синхронизируемый кварцевым резонатором, тактовая частота повышена до 3 МГц; мультиплексированную во времени шину адреса/данных (ША/Д). Освободившиеся в результате выводы использованы для трех маскируемых сигналов прерывания по вектору и одного немаскируемого, двух линий последовательного ввода и вывода, а также выходного сигнала начальной установки. Система команд К1821ВМ85А включает систему команд КР580ВМ80 и две дополнительные команды последовательного ввода и вывода (эти две команды позволяют также оперировать с маской прерывания).

Назначение выводов МП К1821ВМ85А приведено в табл. 2.6.

Архитектурные особенности МП К1821ВМ85А, связанные с построением внутрисхемного эмулятора, аналогичны особенностям МП КР580ВМ80.

Таблица 2.6

Номер вывода	Обозначение вывода	Назначение вывода
1, 2	X1, X2	Выводы для подключения кварцевого резонатора или *RC*-цепи
3	RESET OUT	Выход начальной установки
4	SOD	Выход последовательных данных
5	SID	Вход последовательных данных
6	TRAP	Немаскируемый запрос прерывания
7, 8, 9	RST 5.5, RST 6.5, RST 7.5	Входы маскируемых запросов прерывания RST 7.5, RST 6.5, RST 5.5
10	INTR	Запрос прерывания
11	INTA	Подтверждение прерывания
12–19	AD0–AD7	Двунаправленная ША/Д
20	GND	Общий
21–28	A8–A15	Шина восьми старших разрядов адреса (ША)
29, 33	S0, S1	Внутреннее состояние МП
30	ALE	Строб адреса
31	$\overline{WR}$	Запись
32	$\overline{RD}$	Считывание
34	IO/$\overline{M}$	Выбор устройства ввода-вывода или памяти
35	READY	Готовность
36	$\overline{RESET\ IN}$	Вход начальной установки
37	CLK	Выход тактового сигнала
38	HLDA	Подтверждение захвата
39	HOLD	Захват
40	U_{SS}	Питание

Назначение и возможности использования сигнала ГОТОВНОСТЬ в целях отладки для МП К1821ВМ85А полностью тождественны МП КР580ВМ80.

Структура внутренних регистров отличается наличием в МП К1821ВМ85А дополнительного регистра маски прерываний, старший разряд которого соответствует биту последовательного ввода-вывода. Значение маски прерываний, так же как и значения всех прочих внутренних регистров, доступны для ввода и вывода через системную магистраль МП.

В части сигналов состояния системной магистрали различие двух МП заключается в том, что в приборе К1821ВМ85А имеются два специальных вывода выходных сигналов состояния микропроцессора S0 и S1, которые позволяют определить тип машинного цикла, выполняемого МП. В частности, если оба сигнала находятся в состоянии логической единицы, то это означает, что МП выполняет цикл выборки первого байта команды (кода операции).

Что касается входов прерываний, а также линий последовательного ввода и вывода, то в рассматриваемом ниже варианте построения блока эмулятора эти сигналы не задействованы и полностью переданы в распоряжение пользователя.

Наличие у МП К1821ВМ85А временно́го мультиплексирования данных и младшего байта адреса вызывает необходимость использования буферов адреса и данных для организации внутренней магистрали блока эмулятора.

Структура блока эмулятора. Рассмотрим структурную схему одного из вариантов блока эмулятора БЭ-821 микропроцессора К1821ВМ85А, представленную на рис. 2.8.

Основные технические характеристики блока эмулятора:

Вид эмуляции	Внутрисхемный
Объем эмулируемой памяти, К байт	16 с возможностью расширения до 64
Максимальное число одновременно задаваемых точек останова по адресу	32 К
Объем памяти следа	1К×32 бит
Отладочные функции:	
возможность отладки программ в реальном масштабе времени	Имеется
условия прерывания эмуляции	По контрольной точке (по адресу), пошаговый и поцикловый режимы, по команде оператора, по обращению к отсутствующей памяти
режим трассировки	Имеется
возможность индикации и модификации содержимого памяти эмулятора или МПУ и внутренних регистров МП	Имеется
возможность картирования адресного пространства МП	Имеется
возможность оперативного изменения конфигурации системы	Имеется
Типы интерфейсов связи:	
связь с БВК	Межмодульный параллельный интерфейс МПИ
связь с МПУ	В соответствии со спецификацией на выводы МП
Максимальная тактовая частота, МГц	3
Напряжение питания, В	5 ± 5%

Интерфейсы блока эмулятора БЭ-821. В структуре блока эмулятора БЭ-821, как и в ранее рассмотренных случаях, можно выделить три интерфейса. Связь блока эмулятора с базовым вычислительным комплексом осуществляется через межмодульный параллельный интерфейс МПИ. Внутренние шины данных и адреса связывают между собой все устройства блока эмулятора (БЭ), за исключением буфера связи с МПУ. Интерфейс связи с МПУ пользователя соответствует спецификации выводов и временны́м диаграммам целевого МП. Особенностью

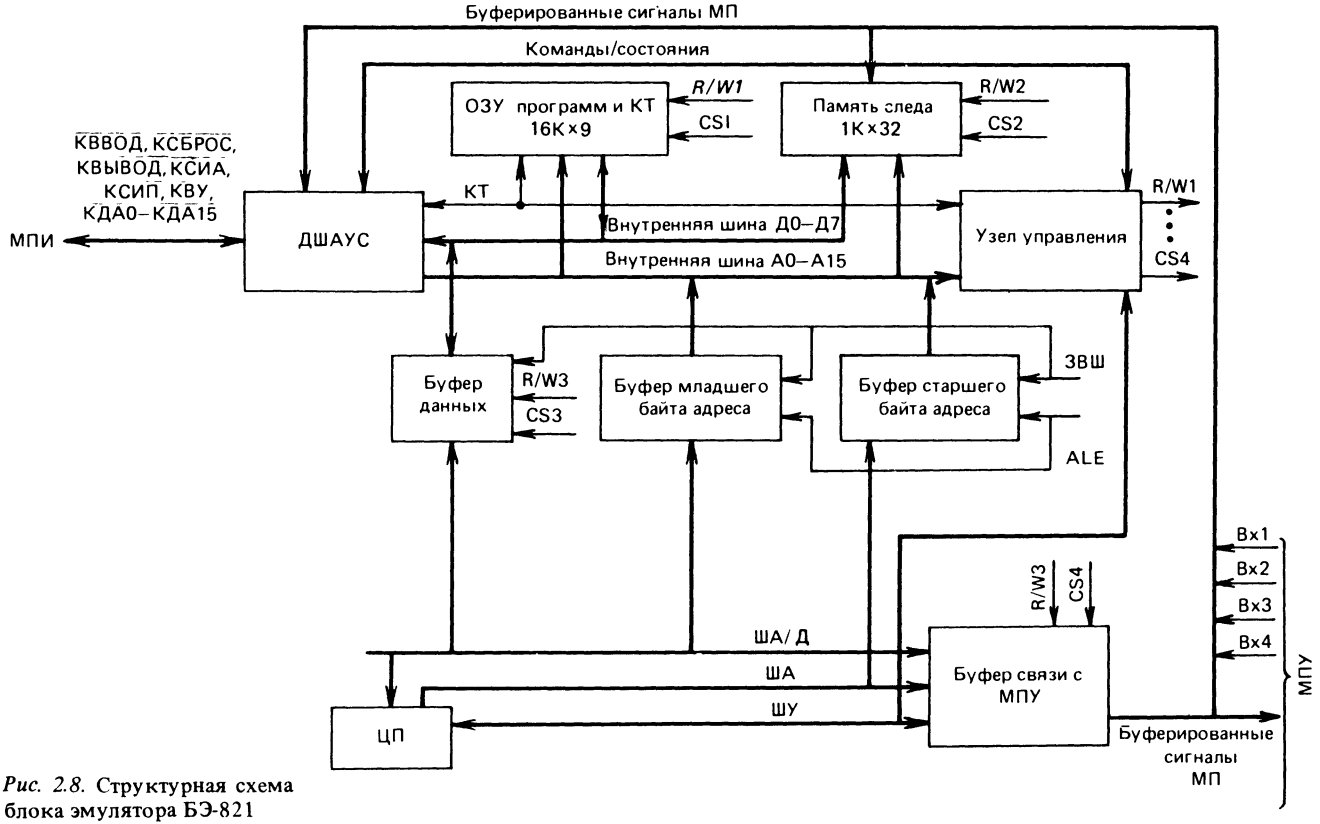


Рис. 2.8. Структурная схема блока эмулятора БЭ-821

интерфейса связи с МПУ является наличие дополнительных входов (зондов) в блоке эмулятора, позволяющих по выбору оператора подключить четыре произвольные точки МПУ к памяти следа блока эмулятора.

Дешифратор адреса и управляющих сигналов (ДШАУС). Обеспечивает связь между БЭ и базовым вычислительным комплексом. Через него осуществляется запись из БВК управляющих слов (команд) в узел управления, а также программ, данных и битов контрольных точек в ОЗУ программ и КТ. Для считывания информации в БВК доступны ОЗУ программ и КТ, память следа, узел управления (слово состояния БЭ), а также буферированные сигналы МП. Коммутацию всех источников информации к ДШАУС выполняет узел управления. Со стороны БВК дешифратор адреса и управляющих сигналов представляется как две адресуемые ячейки памяти: регистр адреса (адрес 165000) и регистр данных (адрес 165002), соответствующих внутренним шинам блока эмулятора. Построен ДШАУС по типовой схеме, обеспечивающей выполнение протокола обмена, предусмотренного для МПИ.

Центральный процессор блока эмулятора (ЦП). Выполняет отлаживаемую программу и обеспечивает вывод содержимого внутренних регистров для последующей индикации. Управляющие сигналы ЦП используются для организации работы узла управления блока эмулятора. В качестве ЦП использован МП типа К1821ВМ85А. Останов МП в служебных целях осуществляется путем подачи соответствующего уровня напряжения на вход ГОТОВНОСТЬ, который вырабатывается узлом управления. Доступ к этому входу предоставляется также и устройству пользователя.

Буфер данных. Обеспечивает доступ МП к внутренней шине данных блока эмулятора и тем самым к ОЗУ программ. Направление передачи данных определяется сигналом R/W3, который вырабатывается узлом управления в соответствии с циклом, выполняемым МП. Сигнал CS3 обеспечивает выборку буфера данных на время обращения МП к ОЗУ программ. Сигнал захвата внутренних шин (ЗВШ) переводит буфер данных в третье состояние.

Буфера адреса. Обеспечивают передачу адреса, генерируемого МП, на внутреннюю шину адреса блока эмулятора. Причем эта передача выполняется независимо от того, обращается МП к памяти эмулятора или к памяти МПУ. Это необходимо для работы памяти КТ (ОЗУ программ и КТ) и карты памяти (узел управления). Сигнал ЗВШ переводит буферы адреса в высокоимпедансное состояние на время доступа БВК к внутренним шинам блока эмулятора. Разделение буфера адреса на буферы младшего и старшего байтов определено архитектурой МП К1821ВМ80А, у которого шина данных и младшего байта адреса мультиплексированы.

Буфер связи с МПУ. Обеспечивает полное буферирование сигналов блока эмулятора по отношению к МПУ, что позволяет работать блоку эмулятора даже при неисправном МПУ. Выходы буфера управля-

ются напряжением питания, поступающим от МПУ. При отсутствии на МПУ питания выходы буфера переводятся в третье состояние с целью защиты схем МПУ. Направление передачи по ШД определяется сигналом R/W3. Буфер связи включается сигналом CS4, который вырабатывается узлом управления, на время обращения МП к МПУ. Отладка аппаратурных средств пользователя в статическом режиме обеспечивается тем, что выходные сигналы МП при его останове „защелкиваются" в буфере связи сигналом узла управления на время выполнения служебных операций. Линии последовательных ввода и вывода данных МП постоянно предоставлены в распоряжение пользователя.

Оперативное ЗУ программ и КТ блока эмулятора. Предназначено для хранения отлаживаемых программ и данных, служебной программы, а также для задания адресов прерывания эмуляции. Все ячейки ОЗУ имеют 9-разрядный формат: восемь разрядов — байт программы и один разряд — бит КТ. Выход разряда контрольной точки подключен к узлу управления, а также с целью контроля доступен и для БВК. В случае если при обращении по очередному адресу бит КТ оказывается установлен, то в узле управления запускается механизм останова МП. Контрольную точку выбирают независимо от того, обращается МП к памяти эмулятора или к памяти МПУ, а также независимо от того, выполняется цикл чтения или записи.

Очевидно, что доступ к ОЗУ программ и КТ возможен как со стороны БВК, так и со стороны МП К1821ВМ85А. Выборка ОЗУ выполняется сигналом CS1, а режим чтения и записи определяется сигналом R/W1. Объем памяти ОЗУ программ составляет 16К байт и может быть расширен до 64К байт установкой дополнительных БИС памяти.

Память следа, или историческая память, представляет собой по существу логический анализатор состояний. Объем памяти следа составляет 1КХ32 бит. Во время каждого цикла МП в память следа записываются: 16 разрядов адреса, 8 разрядов данных, 4 разряда сигналов управления (INTA, S0, S1, IO/M) и 4 разряда сигналов зондов, подключаемых в произвольных точках МПУ. Запись в память следа прекращается в момент прерывания эмуляции. В результате в памяти следа хранится информация о последних 1К циклах МП. В память следа по команде оператора информация может поступать или со стороны внутренних шин данных и адреса блока эмулятора, или со стороны буферированных сигналов интерфейса МПУ. Выбирается память следа сигналом CS2, а режим чтения или записи задается сигналом R/W2.

Узел управления. Основное назначение узла управления заключается в выполнении всех необходимых коммутаций в блоке эмулятора. Узел управления вырабатывает все управляющие сигналы блока эмулятора, указанные на схеме (рис. 2.8): осуществляет подключение к шинам эмулятора соответствующих блоков, определяет направление передачи информации через буферы, управляет остановом и запуском МП К1821ВМ85А.

Инициализация узла управления осуществляется сигналом СБРОС канала МПИ, сигналами контрольных точек из ОЗУ программ КТ и в основном командами оператора, поступающими из БВК. Со стороны БВК узел управления представляется двумя регистрами, адресуемыми как ячейки памяти: регистр режимов и команд (адрес 165006 на запись) и регистр состояний (адрес 165004 на чтение). Каждый бит регистра режимов и команд обеспечивает подключение к шинам эмулятора соответствующих источников и приемников информации или указывает направление передачи данных.

Узел управления обеспечивает доступ к внутренним шинам эмулятора попеременно или МП, или базовому вычислительному комплексу. Первый случай имеет место, когда МП выполняет программу. Выходы регистров адреса и данных ДШАУС в это время находятся в третьем состоянии. Второй случай имеет место, когда МП остановлен и БВК через ДШАУС получает возможность обратиться к внутренним шинам адреса и данных блока эмулятора. Буферы данных, младшего и старшего байтов адреса при этом переводятся в третье состояние. Во время работы МП узел управления обеспечивает поступление в МП команд и данных или из ОЗУ программ и КТ, или из памяти МПУ пользователя в соответствии с заданной оператором картой памяти. Естественно, что во время выполнения служебной программы МП К1821ВМ85А подключается к памяти блока эмулятора независимо от назначенной карты памяти.

Во время останова МП узел управления по команде оператора обеспечивает доступ БВК через ДШАУС или к ОЗУ программ и КТ, или к памяти следа, или к буферированным сигналам МП К1821ВМ85А.

Во время работы МП узел управления по команде оператора осуществляет подключение к входам памяти следа или внутренней шины эмулятора, или буферированных сигналов МП.

Узел управления обеспечивает подключение к МП К1821ВМ85А внутреннего кварцевого резонатора или внешних цепей синхронизации.

Узел управления осуществляет также начальный запуск и останов МП. Сигнал останова вырабатывается по одному из четырех условий: выполнение очередной команды в пошаговом режиме или цикла в поцикловом, появление КТ, команда оператора на останов и обращение МП по адресу, объявленному в карте памяти как отсутствующий.

Алгоритм функционирования блока эмулятора. Вывод содержимого внутренних регистров МП и их модификация производятся микропроцессором при выполнении служебной программы. Учитывая, что МП КР580ВМ80 и К1821ВМ85А практически совпадают по составу внутренних регистров и системе команд, служебные программы и алгоритмы их выполнения аналогичны. Так же как и в блоке эмулятора БЭ-580, для сокращения аппаратурных затрат служебная программа после останова МП записывается в ОЗУ программ и КТ из БВК. Переход на ее выполнение осуществляется подстановкой команды безусловного перехода. Содержимое соответствующих областей ОЗУ программ блока эмулятора на

время выполнения служебной программы запоминается БВК с последующим восстановлением.

Запуск МП на выполнение программы с заданного адреса осуществляется по команде оператора, поступающей из БВК в узел управления. Узел управления обеспечивает загрузку служебной программы в блок эмулятора и начальную установку МП. По служебной программе происходит загрузка внутренних регистров МП, в том числе счетчика команд, т. е. обеспечивается запуск с необходимого адреса.

В процессе выполнения программы пользователя адрес и сигнал выборки поступают на разряд контрольных точек ОЗУ программ и КТ независимо от того, с какой памятью – эмулятора или МПУ пользователя – работает МП. Появление КТ, так же как и выполнение любого другого условия останова, приводит к установке в узле управления триггера, запрещающего поступление на МП сигнала ГОТОВНОСТЬ.

После подачи команды на пуск МП базовый вычислительный комплекс в соответствии с управляющей программой определяет его состояние (выполняет МП программу или же остановлен). Анализируется состояние МП путем многократной проверки сигнала ALE. Если сигнал ALE периодически вырабатывается, то БВК индицирует сообщение *Идет выполнение программы* и переходит на опрос клавиатуры для получения очередной команды оператора. Если же сигнал ALE длительное время (порядка 0,5 с) не вырабатывается, то это рассматривается как свидетельство об останове МП. В этом случае БВК анализирует причину останова, которой могут служить: выполнение заданного оператором условия останова, длительное отсутствие сигнала ГОТОВНОСТЬ от МПУ, поступление от МПУ сигнала требования прямого доступа к памяти. В любом случае БВК индицирует соответствующее сообщение.

Управление обращением МП к памяти эмулятора или МПУ пользователя выполняется узлом управления в реальном масштабе времени в соответствии с картой памяти, задаваемой оператором. Схемотехнически картирование выполнено на базе БИС статического ОЗУ объемом 4КХ Х1 бит. Содержимое этого ОЗУ задается во время формирования карты памяти. В процессе выполнения программы пользователя старшие 12 разрядов адреса поступают на входы ОЗУ карты памяти и в зависимости от бита, хранящегося по данному адресу, происходит обращение к памяти программ блока эмулятора или МПУ пользователя. Интервал картирования равен 20_8. Аналогичным образом некоторые области адресного пространства могут быть объявлены как отсутствующие в архитектуре МПУ. Схемотехнически это обеспечивается дополнительной БИС статического ОЗУ. Обращение МП к отсутствующей памяти вызывает прерывание процесса эмуляции, переход МП на выполнение служебной программы и соответствующую индикацию на экране дисплея. Отметим, что индикация содержимого памяти на дисплее БВК выполняется в соответствии с картой памяти.

Информацию в память следа по желанию оператора записывают как в каждом машинном цикле МП, так и по появлению КТ. В последнем случае при появлении КТ останов МП и индикация не происходят.

В отличие от блока эмулятора БЭ-580 дизассемблирование выполняется с использованием информации, записанной в памяти следа, а не считываемой с внутренней ШД в момент останова МП. Это позволяет выполнять дизассемблирование не только в пошаговом или поцикловом режимах, но и при останове по КТ или команде оператора.

Отладка аппаратурных средств в статическом режиме обеспечивается тем, что выходные сигналы МП при его останове защелкиваются в буфере связи с МПУ и сохраняют свое значение на время выполнения служебных операций вплоть до получения команды на продолжение эмуляции.

Диагностика выходных шин МП в отлаживаемом МПУ проводится путем сличения сигналов внутренних шин блока эмулятора с буферированными сигналами МП. Для проверки МП К1821ВМ85А автоматически останавливается после выполнения каждого машинного цикла. При расхождении уровней сравниваемых сигналов БВК выводит на дисплей информацию об ошибках. Аналогично может быть проведена проверка и входных шин, но для этого необходима разработка специальных тестов пользователя.

Блок эмулятора практически не налагает ограничений на использование аппаратурных или программных ресурсов МП. Работа блока эмулятора осуществляется под управлением программы, аналогичной DEB580 (см. §3.1).

Глава 3

ТЕХНОЛОГИЯ ПРОГРАММНО-АППАРАТУРНОЙ ОТЛАДКИ МПУ

3.1. Отладка МПУ на базе микропроцессора КР580ВМ80

Рассмотрим более подробно действия оператора УОК МИКРОСОТ в процессе комплексной отладки программно-аппаратурных средств МПУ, построенных на базе микропроцессора КР580ВМ80. Как уже указывалось ранее, в состав аппаратурных средств комплекса в этом случае включаются БВК типа ДВК-2М и блок эмулятора БЭ-580. Работа комплекса осуществляется под управлением программы DEB580. В качестве БВК может быть использована и другая микроЭВМ с аналогичной системой команд и операционной системой. Для работы программы DEB580 достаточна конфигурация комплекса, включающая центральный процессор, алфавитно-цифровой дисплей, устройство внешней памяти на гибких магнитных дисках типа НГМД-6022, блок эмулятора БЭ-580.

Управляющая программа DEB580 поддерживает выполнение комплексом следующих функций: загрузка отлаживаемой программы в кодах МП с НГМД или клавиатуры в ОЗУ блока эмулятора или МПУ; на-

чальная установка содержимого всех рабочих регистров МП; запуск МП на выполнение программы с произвольного адреса; выполнение программы пользователя в реальном масштабе времени; останов МП по команде оператора или по контрольным точкам в заданных адресах; выполнение программы пользователя по шагам или циклам; вывод на устройство отображения и модификация содержимого внутренних рабочих регистров МП; вывод на устройство отображения содержимого памяти блока эмулятора и МПУ; модификация содержимого памяти блока эмулятора и ОЗУ МПУ; циклическое выполнение заданной части программы пользователя; задание конфигурации АС блока эмулятора и МПУ; полное управление блоком эмулятора с клавиатуры БВК.

Входными данными для программы DEB580 являются команды оператора и файл отлаживаемой программы, либо полученный в результате работы пакета программ КОП-580, либо загружаемый оператором с клавиатуры непосредственно в кодах МП.

Выходными данными являются сообщения оператору.

Для обращения к программе файл DEB580 SAV формата отображения памяти, хранящейся на системном устройстве (SY), загружается в память и запускается по команде монитора ОС ДВК:

R DEB580 <ВК>

После запуска программы DEB580 происходит очистка экрана и начинается диалог программы с оператором. В ходе диалога оператору задаются вопросы и предлагаются возможные варианты ответов, каждому из которых соответствует цифра 0 или 1. Ввод оператором с клавиатуры той или иной цифры означает выбор одного из предлагаемых ответов. Каждый ответ оператора завершается нажатием клавиши <ВК>. На протяжении всей работы программой выдаются информационные сообщения или сообщения об ошибках, допущенных оператором. Пример начального диалога программы DEB580 представлен ниже.

СХЕМНЫЙ ЭМУЛЯТОР КР580ВМ80

В КАКОЙ СИСТЕМЕ СЧИСЛЕНИЯ ВЫ ХОТИТЕ РАБОТАТЬ?
0 – ВОСЬМЕРИЧНАЯ СИСТЕМА
1 – ШЕСТНАДЦАТЕРИЧНАЯ СИСТЕМА – 1 <ВК>
ВЫБРАНА ШЕСТНАДЦАТЕРИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ
ОБ ЬЕМ ПАМЯТИ ЭМУЛЯТОРА: 0 – 2FFF
ЗАДАЙТЕ НАЧАЛЬНЫЙ АДРЕС ПЕРЕМЕЩАЕМОГО БЛОКА ОЗУ ЭМУЛЯТОРА – 8000 <ВК>
ОЧИСТКА ПАМЯТИ ЭМУЛЯТОРА – ОЧИСТКА ЗАКОНЧЕНА
ОПРЕДЕЛИТЕ КОНФИГУРАЦИЮ СИСТЕМЫ
0 – ПАМЯТЬ ПОЛЬЗОВАТЕЛЯ, 1 – ПАМЯТЬ ЭМУЛЯТОРА – ?1 <ВК>
0 – ВНЕШНИЙ ГЕНЕРАТОР, 1 – ГЕНЕРАТОР ЭМУЛЯТОРА – ?0 <ВК>
С КАКОГО УСТРОЙСТВА БУДЕТ ЗАГРУЖЕНА ПРОГРАММА? DX: <ВК>
НАЗНАЧЕНИЕ РЕЖИМА ЭМУЛЯЦИИ
0 – НЕПРЕРЫВНЫЙ
1 – ПОШАГОВЫЙ – 1 <ВК>

ВВЕДИТЕ ИМЯ ПРОГРАММЫ ПОЛЬЗОВАТЕЛЯ – T580 <ВК>
АДРЕС ЗАГРУЗКИ = 10
ДЛИНА МОДУЛЯ = 16

В начале диалога оператору предлагается выбрать привычную для него систему счисления: восьмеричную или шестнадцатеричную. В дальнейшем вся индикация и ввод информации будут осуществляться в выбранной системе счисления. Изменение системы счисления в процессе работы программы не допускается.

Программа DEB580 подтверждает выбор системы счисления и автоматически переходит к выявлению объема ОЗУ программ и КТ блока эмулятора. Как уже отмечалось ранее, в распоряжении пользователя может быть ОЗУП объемом 16—32К байт. Результат проверки индицируется на устройстве отображения (без учета объема перемещаемого банка ОЗУП).

Далее оператору предлагается задать начальный адрес перемещаемого банка ОЗУП блока эмулятора. Начальный адрес задается с дискретностью 4К и в шестнадцатеричной системе счисления выбирается из ряда: 3000, 4000 5000, . . ., F000.

Затем БВК под управлением программы DEB580 переходит к очистке ОЗУП и КТ блока эмулятора (запись в память нулей) и проверке очистки. Результат проверки — сообщение оператору *Очистка закончена.* Если указанное сообщение не появляется, это означает, что память блока эмулятора неисправна и работа с системой невозможна. Необходимо устранить неисправность и перезагрузить программу.

Далее оператор определяет конфигурацию системы: задает память МПУ или блока эмулятора, внешний или внутренний тактовый генератор. Из всех возможных сочетаний вариант использования внутреннего генератора и памяти МПУ запрещен. Причина заключается в отсутствии сигнала строба записи состояния (сигнала тактового генератора TSTB) в интерфейсе связи блока эмулятора с МПУ. При работе с памятью блока эмулятора обращение к портам ввода-вывода МПУ не запрещено.

На вопрос управляющей программы: *С какого устройства будет загружена программа?* действия оператора аналогичны вводу команды BOOT (см. ниже). Ввод программы пользователя возможен с клавиатуры или НГМД. Программа на ГМД должна быть предварительно получена с помощью кросс-ассемблера, входящего в состав КОП-580. Если при загрузке программы пользователя в качестве устройства ввода назначено устройство, отличное от „TT:" или „DX:", индицируется сообщение *Такого устройства нет!.* Одновременно с вводом программы по выбору оператора назначается непрерывный или пошаговый режим работы блока эмулятора. В дальнейшем режим может быть изменен соответствующей командой. Оканчивается ввод программы пользователя с НГМД сообщением об адресе загрузки и длине загруженного модуля в байтах. По завершении начального диалога на экране дисплея индицируется символ *. Этот символ всегда означает предложение оператору ввести следующую команду.

Программой DEB580 предусмотрено выполнение 18 команд оператора по управлению работой блока эмулятора БЭ-580. Ниже приведены описания и особенности выполнения каждой команды.

BOOT. По команде BOOT осуществляется загрузка отлаживаемой программы в кодах микропроцессора с НГМД или клавиатуры в ОЗУП блока эмулятора.

В ответ на запрос:

С какого устройства будет загружена программа?

оператор должен ввести:

DX: – если программа будет загружена с ГМД

или

TT: – если программа будет загружаться с клавиатуры.

Пример. При ответе оператора „TT:" индицируется сообщение:

```
ВВЕДИТЕ КОДЫ КОМАНД ЧЕРЕЗ <ВК>
КОНЕЦ ВВОДА – 777В ИЛИ IFFH
ЗАДАЙТЕ НАЧАЛЬНЫЙ АДРЕС – 20<ВК>
20В/61 <ВК>
21В/200 <ВК>
22В/ 0 <ВК>
23В/777 <ВК>
*
```

В результате в ячейки с адресами 20, 21, 22 загружены коды 61, 200, 0.

При ответе оператора „DX:" индицируется сообщение:

```
НАЗНАЧЕНИЕ РЕЖИМА ЭМУЛЯЦИИ
0 – НЕПРЕРЫВНЫЙ
1 – ПОШАГОВЫЙ – 1 <ВК>
ВВЕДИТЕ ИМЯ ПРОГРАММЫ ПОЛЬЗОВАТЕЛЯ – DX: T580 <ВК>
АДРЕС ЗАГРУЗКИ = 20В
ДЛИНА МОДУЛЯ = 26В
```

Здесь и далее индексы В и Н означают соответственно восьмеричную и шестнадцатеричную систему счисления.

Отметим, что с НГМД загружаются файлы, полученные с помощью КОП-580 и хранящиеся в формате OBJ или SAV. Тип файла по умолчанию: OBJ

МЕМ. По команде МЕМ можно изменить содержимое любой ячейки памяти эмулятора, адрес которой задается оператором. Во избежание случайной замены данных выдается контрольное сообщение.

Пример.

```
* МЕМ <ВК>
АДРЕС ЯЧЕЙКИ ПАМЯТИ? 100 : ВК>
100 = . 0>
ХОТИТЕ ИЗМЕНИТЬ СОДЕРЖИМОЕ ПАМЯТИ (ОТВЕТ ДОЛЖЕН БЫТЬ:
0 – НЕТ, 1 – ДА) ? 1 : ВК>
ВВЕДИТЕ НОВОЕ СОДЕРЖИМОЕ – 3 <ВК>
*
```

После выполнения этой операции в ячейке 100 содержится 3.

R. По команде R происходит изменение содержимого РОН МП. После подачи команды на экране индицируется:

Имя РОН, содержимое которого Вы хотите изменить?.

Оператор должен ввести имя одного из регистров ЦП: *A, B, C, D, E, H, L* или *SP*. Затем индицируется старое содержимое РОН, и оператор должен ввести новое содержимое. Если процесс эмуляции был остановлен не в первом машинном цикле выполняемой команды, индицируется сообщение:

Изменить содержимое РОН невозможно!

Последнее связано с тем, что изменение содержимого регистров МП выполняется с помощью служебной программы. Очевидно, что ее выполнение возможно только после полного завершения текущей команды программы пользователя.

Пример.

```
* R <ВК>
ИМЯ РОН, СОДЕРЖИМОЕ КОТОРОГО ВЫ ХОТИТЕ ИЗМЕНИТЬ? A<ВК>
A = <0> I <ВК>
*
```

После выполнения операции аккумулятор будет содержать 1.

Если при выполнении команды R введено имя регистра, отличное от *A, B, C, D, E, H, L, SP*, то индицируется следующее сообщение:

Такого РОН в ЦП КР580ВМ80 нет!

Оператору необходимо правильно повторить команду.

LOAD. По команде LOAD осуществляется загрузка ОЗУ МПУ. При этом содержимое ячеек памяти блока эмулятора, задаваемых оператором, загружается в соответствующие адреса ОЗУ МПУ.

Пример.

```
* LOAD <ВК>
НАЧАЛЬНЫЙ АДРЕС ЗАГРУЖАЕМОГО МАССИВА ДАННЫХ? 100 <ВК>
КОНЕЧНЫЙ АДРЕС ЗАГРУЖАЕМОГО МАССИВА ДАННЫХ? 200 <ВК>
ЗАГРУЗКА ПАМЯТИ МПУ ЗАВЕРШЕНА.
*
```

Следует обратить внимание, что загрузка ОЗУ МПУ выполняется МП блока эмулятора по служебной программе путем перезаписи содержимого соответствующих ячеек ОЗУП блока эмулятора в ОЗУ МПУ. Предварительно необходимая информация должна быть занесена в ОЗУП эмулятора с помощью, например, команды BOOT. Выполнение команды возможно только после полного завершения текущей команды программы пользователя.

После загрузки кодов в ОЗУ МПУ проводится сравнение записываемых и считываемых затем данных. При отрицательном результате проверки индицируется диагностическое сообщение:

При загрузке памяти МПУ произошел сбой.

Повторите загрузку!

TYPE. По команде TYPE индицируется содержимое памяти эмулятора или памяти МПУ (в зависимости от того, какой тип памяти используется в системе в текущий момент времени), начиная с адреса, заданного оператором. Содержимое памяти индицируется в виде таблицы. Наличие контрольной точки в соответствующей ячейке индицируется символом *. Пример выполнения команды TYPE приведен ниже:

```
* TYPE
НАЧАЛЬНЫЙ АДРЕС? 0
СОДЕРЖИМОЕ ПАМЯТИ МПУ
АДРЕС  0  1  2  3  4 5 6  7  8  9 A  B  C  D E  F
0000/31 C7 23 00 00 00 00 00 F3 E5 00 46 00 C3 46 01
. . . . . .
. . . . . .
0090/4 F06 00 21 CF03 09 7E 32 F6 23 6F 26 00 29 01
00A0/06 03 09 0E 03 E7 CD CE 02 CD 10 02 06 82 CD 48
. . . . . .
. . . . . .
0110/10 CA 0A 02 FE 11 CA 26 01 CD 3E 02 D2 32 02 EB
ВЫВОД СОДЕРЖИМОГО РОН НЕВОЗМОЖЕН
ШИНА АДРЕСА = 10 ШИНА ДАННЫХ = 0 ДЕЙСТВИЕ ЦП: СЧИТЫВАНИЕ
ПАМЯТИ
```

CONF. По команде CONF изменяется конфигурация аппаратурных средств системы.

Пример.

```
* CONF <ВК>
ОПРЕДЕЛИТЕ КОНФИГУРАЦИЮ СИСТЕМЫ.
0 – ПАМЯТЬ ПОЛЬЗОВАТЕЛЯ, 1 – ПАМЯТЬ ЭМУЛЯТОРА – ?1 <ВК>
0 – ВНЕШНИЙ ГЕНЕРАТОР, 1 – ГЕНЕРАТОР ЭМУЛЯТОРА – ?1 <ВК>
ВХОДИТ ЛИ МПУ В КОНФИГУРАЦИЮ СИСТЕМЫ (0 – НЕТ, 1 – ДА)?
0 <ВК>
*
```

Возможные сочетания АС системы соответствуют рассмотренным в начальном диалоге программы DEB580.

STEP. Команда STEP устанавливает пошаговый режим работы блока эмулятора. После выполнения каждой команды отлаживаемой программы в пошаговом режиме осуществляется индикация содержимого внутренних регистров МП и мнемоники выполненной команды.

Пример.

```
* STEP <ВК>
A=<0  B=<0> C=<0> D=<0> E=<0> H=<0> L=<0>  SP=<0>
PC= <13>
БИТЫ УСЛОВИЙ ЦП: C = 0  P = 0  AC = 0  Z = 0  S = 0
* P
ВЫПОЛНЕНА КОМАНДА: LXI H, 40
A=<0> B=<0> C=<0> D=<0> E=<0> H=<0> L=<40> SP=<0>
PC=<16>
БИТЫ УСЛОВИЙ ЦП: C=0 P=0 AC=0 Z=0 S=0
*
```

CR. По команде CR назначается поцикловый режим работы блока эмулятора. Индикация состояния МП осуществляется после выполнения каждого машинного цикла. Если останов МП произошел при вводе первого байта команды, то индикация выполняется аналогично команде STEP. Пример индикации для случая обращения не к первому байту команды приведен ниже:

```
ВЫВОД СОДЕРЖИМОГО РОН НЕВОЗМОЖЕН.
ШИНА АДРЕСА = 6 ШИНА ДАННЫХ = C5
ДЕЙСТВИЕ ЦП: СЧИТЫВАНИЕ ПАМЯТИ
*
```

КТ. По команде КТ оператор может назначить контрольные точки в любом месте адресного пространства МП. Число КТ практически ограничено только объемом ОЗУ КТ блока эмулятора. Адреса точек останова вводятся оператором по запросу программы. Признак окончания ввода контрольных точек – ввод адреса 0.

Пример.

```
* КТ <ВК>
ВВЕДИТЕ АДРЕСА, В КОТОРЫЕ ВЫ ХОТИТЕ УСТАНОВИТЬ КОНТРОЛЬНЫЕ
ТОЧКИ. ПРИЗНАК ОКОНЧАНИЯ ВВОДА КОНТРОЛЬНЫХ ТОЧЕК – ОВ
ИЛИ ОН.
АДРЕС  КОНТРОЛЬНОЙ ТОЧКИ? 20 <ВК>
АДРЕС КОНТРОЛЬНОЙ ТОЧКИ? 33 <ВК>
АДРЕС КОНТРОЛЬНОЙ ТОЧКИ? 0 <ВК>
*
```

При этом контрольные точки останова расставлены в адресах 20 и 33.

Отметим, что по команде КТ невозможно установить контрольную точку в нулевой адрес. Если возникает такая необходимость, можно использовать команду МЕМ, введя число, содержащее единицу в 9-м разряде.

HALT. По команде HALT происходит останов процессора при обнаружении заданного типа машинного цикла.

Пример.

```
* HALT <ВК>
ВВЕДИТЕ КОД СЛОВА СОСТОЯНИЯ
1 – ВЫБОРКА КОДА ОПЕРАЦИИ
2 – ЧТЕНИЕ ПАМЯТИ
3 – ЗАПИСЬ В ПАМЯТЬ
4 – ЧТЕНИЕ СТЕКА
5 – ЗАПИСЬ В СТЕК
6 – ВВОД ОТ ВНЕШНЕГО УСТРОЙСТВА
7 – ВЫВОД НА ВНЕШНЕЕ УСТРОЙСТВО
8 – ОБРАБОТКА ПРЕРЫВАНИЯ
9 – ПОДТВЕРЖДЕНИЕ ОСТАНОВА
10 – ПОДТВЕРЖДЕНИЕ ПРЕРЫВАНИЯ ПРИ ОСТАНОВЕ – 5 <ВК>
ШИНА АДРЕСА = 177B ШИНА ДАННЫХ = 0B
ДЕЙСТВИЕ ЦП: ЗАПИСЬ В СТЕК
*
```

После назначения команды HALT выполнение отлаживаемой программы без участия оператора осуществляется в поцикловом режиме, т. е. не в реальном масштабе времени. При выполнении каждого машинного цикла БВК анализирует его тип путем дешифрации слова состояния МП. Если при выполнении команды HALT за 32 767 циклов не обнаружено условие останова программы, то индицируется сообщение:

Условие останова не найдено.

Для продолжения поиска необходимо повторить команду.

TRACE. По команде TRACE происходит индикация 130 точек трассы программы (значений шины адреса). Выполнение отлаживаемой программы осуществляется в заранее заданном режиме (пошаговом и т. п.). Содержание индикации также определяется заданным режимом. В поцикловом режиме индицируются последовательные значения шины адреса, в пошаговом – значения шины адреса в моменты обращения за первым байтом команды, а в режиме КТ – значения шины адреса в момент обнаружения КТ, в непрерывном режиме индикация не происходит, а выполнение команды TRACE аналогично выполнению команды GO (см. ниже). Пример выполнения команды TRACE представлен ниже:

```
* TRACE
ВВЕДИТЕ АДРЕС ПУСКА ПРОГРАММЫ – 10
РАЗРЕШЕНА РАБОТА С ПРЕРЫВАНИЯМИ (0 – НЕТ, 1 – ДА)? 1
13 16 17 19 IA IB IE IF 20 10
13 16 17 19 IA IB IE IF 20 10
13 16 17 19 IA IB IE IF 20 10
13 16 17 19 IA IB IE IF 20 10
13 16 17 19 IA IB IE IF 20 10
13 16 17 19 IA IB IE IF 20 10
13 16 17 19 IA IB IE IF 20 10
13 16 17 19 IA IB IE IF 20 10
13 16 17 19 IA IB IE IF 20 10
13 16 17 19 IA IB IE IF 20 10
13 16 17 19 IA IB IE IF 20 10
13 16 17 19 IA IB IE IF 20 10
13 16 17 19 IA IB IE IF 20 10
A=<F> B=<40> C=<0> D=<0> E=<40> H=<0> L=<45> SP=<7C>
PC=<28>
БИТЫ УСЛОВИЙ ЦП: C=1 P=1 AC=1 Z=0 S=1
```

Изменение значений счетчика команд происходит по строкам. Если 130 точек трассы недостаточно, необходимо еще раз выполнить команду TRACE с адреса останова программы.

NONSTOP. По команде NONS отменяются поцикловый, пошаговый режим и КТ; тем самым назначается непрерывный режим работы эмулятора.

RES. По команде RES на МП подается сигнал УСТАНОВКА, также обнуляется содержимое регистров общего назначения, указателя стека и битов условий МП.

CYKL. По команде CYKL происходит циклическое выполнение заданного участка программы пользователя, хранящейся в ОЗУП блока эмулятора. В диалоге с оператором запрашивается начальный и конечный адрес зацикливаемого участка программы и выдается сообщение:

Дайте команду на запуск с начального адреса.

Пример.

```
* CYKL <ВК>
АДРЕС ПЕРВОЙ ВЫПОЛНЯЕМОЙ КОМАНДЫ – ?20 <ВК>
АДРЕС ПОСЛЕДНЕГО БАЙТА ЗАЦИКЛИВАЕМОГО УЧАСТКА – ?27 <ВК>
ДАЙТЕ КОМАНДУ НА ЗАПУСК С НАЧАЛЬНОГО АДРЕСА
*
```

Использование команды CYKL удобно для наблюдения временных диаграмм сигналов с помощью осциллографа. При этом следует иметь в виду, что зацикливаемый участок программы дополняется командой безусловного перехода на начало участка.

GO. По команде GO осуществляется запуск программы пользователя с произвольного адреса. После подачи команды GO на экране индицируется сообщение:

Введите адрес пуска программы

Оператор должен ввести начальный адрес программы и <ВК>.

Программу можно запустить с произвольного адреса, с целью автоматического выполнения команд EI или DI индицируется запрос:

Разрешена работа с прерываниями (0 – нет, 1 – да)?

Оператор должен ввести 0 для выполнения команды DI или 1 – команды EI.

Пример.

```
* GO <ВК>
ВВЕДИТЕ АДРЕС ПУСКА ПРОГРАММЫ – 20 <ВК>
РАЗРЕШЕНА РАБОТА С ПРЕРЫВАНИЯМИ (0 – НЕТ, 1 – ДА)? 0 <ВК>
ВЫПОЛНЕНИЕ ПРОГРАММЫ . . .
*
```

P. По команде P продолжается выполнение программы с текущего адреса (определяемого значением счетчика команд или шиной адреса) в текущем режиме.

STOP. По команде STOP происходит останов выполнения программы пользователя, выполняемой в непрерывном режиме. При поступлении этой команды эмуляция прекращается и индицируется рабочая информация, доступная в данный момент.

SHOW. По команде SHOW индицируется конфигурация технических средств системы: память эмулятора или МПУ, генератор эмулятора или МПУ, а также информация о текущем режиме эмуляции и состоянии МП в текущий момент времени.

Пример.

```
* SHOW <ВК> .
ПАМЯТЬ ЭМУЛЯТОРА
ГЕНЕРАТОР МПУ
НЕПРЕРЫВНЫЙ РЕЖИМ
ВЫПОЛНЕНИЕ ПРОГРАММЫ . . .
*
```

В большинстве случаев по окончании выполнения команды индицируется информация о состоянии МП в текущий момент времени по образцу, представленному выше для команд STEP и CR.

При работе с программой DEB580 оператору дается ряд информационных сообщений или сообщений об ошибках:

ЦП в захвате

Сообщение выдается в случае, когда МП находится в состоянии захвата.

Останов по неготовности МПУ.

Сообщение выдается в случае, когда на блок эмулятора в течение примерно 500 мкс поступает от МПУ сигнал неготовности.

Ответ неверен, повторите!

Сообщение выдается тогда, когда оператор случайно выбрал неправильный номер режима.

Не восьмеричное число! Повторите ввод!

Сообщение выдается тогда, когда оператор ввел символ, отличный от 0 – 7 при работе в восьмеричной системе счисления.

Не шестнадцатеричное число! Повторите ввод!

Сообщение выдается в случае, если оператор ввел символ, отличный от 0–9 или A–F при работе в шестнадцатеричной системе счисления.

Введите число!

Сообщение выдается тогда, когда в ответ на запрос о вводе адреса или данных оператор ограничился вводом символа <ВК>.

Команды XXXX не существует!

Сообщение выдается в случае, если оператор ввел неправильную команду.

Память отключена!

Сообщение выдается в случае, если при загрузке программы DEB580 обнаружен аппаратурный сбой.

Общая последовательность действий при проведении отладки ПС и АС МПУ была рассмотрена в §1.8. Число возможных неисправностей или ошибок при разработке МПУ, а тем более число возможных приемов их обнаружения весьма велико. Непосредственные действия оператора в каждом конкретном случае зависят как от архитектуры МПУ, характера неисправности или отыскиваемый ошибки, так и от опыта оператора, его склонности к тем или иным способам ведения отладки.

Рассмотрим несколько возможных способов обнаружения наиболее часто встречающихся неисправностей.

Пример 1. Весьма распространенным типом неисправности МПУ, построенного на базе МП КР580ВМ80, является неустойчивая работа тактового генератора. Причем сбои в его работе могут быть достаточно редкими и обнаружить их только с помощью осциллографа не удается. Для обнаружения такого рода неисправности весьма эффективно использование команды TRACE.

Первоначально отлаживаемая программа или ее фрагмент загружается в ОЗУП блока эмулятора, и в пошаговом режиме выполняется команда TRACE. Предварительно должен быть назначен внутренний генератор блока эмулятора. Полученная в результате трасса рассматривается как эталонная (см. порядок выполнения команды TRACE). Затем с помощью команды CONF назначается внешний тактовый генератор и несколько раз подряд выполняется команда TRACE. Как правило, в случае неустойчивой работы тактового генератора МПУ получаемая трасса лишь на нескольких первых шагах совпадает с эталонной, а далее счетчик команд начинает принимать трудно объяснимые случайные значения, в большинстве случаев расположенные в старшей области адресного пространства. Иногда процессор может даже вернуться на выполнение заданной программы. Момент ухода от эталонной трассы носит случайный характер. Пример выполнения команды TRACE при неустойчиво работающем тактовом генераторе МПУ представлен ниже:

```
* TRAC
ВВЕДИТЕ АДРЕС ПУСКА ПРОГРАММЫ – 10 <ВК>
РАЗРЕШЕНА РАБОТА С ПРЕРЫВАНИЯМИ (0 – НЕТ, 1 – ДА)?1 <ВК>
13 16 17 19 IA IB IE IF 20 10
13 16 17 19 IA IB IE IF 20 10
13 16 17 19 IA IB IE IF 20 10
13 16 17 19 FFFE FFFC FFFA FFF8 FFF6 FFF4
FFF2 FFF0 FFEE FFEC FFEA FFE8 FFF6  FFF4   FFF2   FFF0
FFCE FFCC FFCA FFC8 FFC6 FFC4 FFC2  FFC0   FFAE   FFAC
FFAA FFA8 FFA6 FFA4 FFA2 16 17 19 IA  IB
IE IF 20 10 13 16 17 19 IA IB
IE IF 20 10 13 16 17 19 IA IB
IE IF 20 10 13 16 17 19 IA IB
IE IF 20 10 13 16 17 19 IA IB
IE IF 20 10 13 16 17 19 IA IB
IE IF 20 10 13 16 17 19 IA IB
A=<F> B=<40> C=<0> D=<0> E=<40> H=<0> L=<45> SP=<7C>
PC=<28>
БИТЫ УСЛОВИЙ ЦП: C=1 P=1 AC=1 Z=0 S=1
```

Для сравнения в качестве эталонной использована ранее приведенная трасса.

Пример 2. Наиболее часто встречающимся видом неисправности МПУ является короткое замыкание или обрыв проводников. Неисправности подобного рода в части цепей связи с памятью МПУ легко выявляются с помощью команды TYPE. Естественно, что предварительно командой CONF должна быть назначена работа с памятью МПУ. Ниже показан ра-

нее рассмотренный пример выполнения команды TYPE, но для случая короткого замыкания линий нулевого и первого разрядов адреса:

```
* TYPE
НАЧАЛЬНЫЙ АДРЕС? 0
        СОДЕРЖИМОЕ ПАМЯТИ МПУ
АДРЕС 0 1 2 3 4 5 6 7 8 9 A B C D E
0000/ 31 31 31 00 00 00 00 00 F3 F3 F3 46 00 00 00 01
. . . . . .
. . . . . .

0090/ 4F 4F 4F 21 CF CF CF 7E 32 32 32 6F 26 26 26 01
00A0/D6 D6 D6 0E 03 03 03 CE 02 02 02 02 06 06 06 48
. . . . . .
. . . . . .
0110/10 10 10 02 FE FE FE 26 01 01 01 02 D2 D2 D2 EB
ВЫВОД СОДЕРЖИМОГО РОН НЕВОЗМОЖЕН.
ШИНА АДРЕСА = 10 ШИНА ДАННЫХ = 0 ДЕЙТСИВЕ ЦП: СЧИТЫВАНИЕ
ПАМЯТИ
*
```

Входной ток выхода ИС в состоянии низкого уровня, как правило, много больше, чем выходной ток выхода в состоянии высокого уровня. При появлении сигнала низкого уровня хотя бы на одном из закороченных выходов ИС значение 0 примут оба разряда. Поэтому значения двух младших двоичных разрядов адреса 00, 01 и 10 в случае их закоротки ошибочно примут одинаковое значение 00 во всех трех случаях и считываемая информация будет одинакова соответственно для каждых трех последовательных ячеек памяти.

При сравнении двух приведенных в данном параграфе распечаток содержимого одной и той же памяти видно, что во второй распечатке содержимое ячеек памяти с адресами 91 и 92 ошибочно совпадает с содержимом ячейки 90; ячеек 95 и 96 – с ячейкой 94 и т. д.

Ниже приведен пример выполнения команды TYPE, но для случая обрыва в цепи нулевого разряда данных:

```
* TYPE
НАЧАЛЬНЫЙ АДРЕС? 0
        СОДЕРЖИМОЕ ПАМЯТИ МПУ
АДРЕС 0 1 2 3 4 5 6 7 8 9 A B C D E F
0000/31 C7 23 01 01 01 01 01 F3 F5 01 47 01 C3 47 01
. . . . . .
. . . . . .

0090/4F 07 01 21 CF 03 09 7F 33 F7 23 6F 27 01 29 01
00A0/D7 03 09 0F 03 E7 CD CF 03 CD 11 03 07 83 CD 49
. . . . . .
. . . . . .
0110/11 CB 0B 03 FF 11 CB 27 01 CD 3F 03 D3 33 03 EB
ВЫВОД СОДЕРЖИМОГО РОН НЕВОЗМОЖЕН.
ШИНА АДРЕСА = 10 ШИНА ДАННЫХ = 0 ДЕЙСТВИЕ ЦП: СЧИТЫВАНИЕ
ПАМЯТИ
*
```

Свободный, ненагруженный вход схем ТТЛ, как правило, воспринимается как высокий уровень, поэтому во всех проиндицированных данных в нулевом разряде присутствует 1.

3.2. Отладка МПУ на базе однокристальной микроЭВМ КМ1816ВЕ48

Состав АС для комплексной программно-аппаратурной отладки МПУ, построенного на базе ОЭВМ КМ1816ВЕ48, аналогичен комплексу для отладки МПУ на базе МП КР580ВМ80. Отличие заключается в использовании блока эмулятора БЭ-816 вместо БЭ-580. Работа комплекса МИКРОСОТ в этом случае осуществляется под управлением программы DEB816. Функциональное назначение программы DEB816, а также порядок обращения к ней аналогичны программе DEB580.

Начальный диалог. После загрузки программы DEB816 в ходе начального диалога осуществляется задание параметров отладочной системы.

Базовый вычислительный комплекс очищает память блока эмулятора и выводит на дисплей следующее сообщение:

```
ЭМУЛЯТОР КР1816
ОЧИСТКА ПАМЯТИ ПРОГРАММ ЭМУЛЯТОРА
ОЧИСТКА ЗАКОНЧЕНА
КОНТРОЛЛЕР РАБОТАЕТ В РЕЖИМЕ ОТЛАДОЧНОГО КРИСТАЛЛА ИЛИ МК ЭВМ?
1 – ОДНОКРИСТАЛЬНАЯ МИКРОЭВМ; 0 – ОТЛАДОЧНЫЙ КРИСТАЛЛ –
```

Как уже отмечалось выше, под режимом однокристальной микроЭВМ подразумевается режим работы с внутренней памятью программ ОЭВМ, т. е. эмуляции БИС КМ1816ВЕ48. Под режимом отладочного кристалла подразумевается режим работы с внешней памятью, т. е. эмуляции БИС КМ1816ВЕ35. Предварительно переключатели П2 и П3 блока эмулятора должны быть установлены соответственно выбираемому режиму.

Нажатием соответствующей клавиши оператор назначает необходимый режим работы. Если выбран режим эмуляции БИС КМ1816ВЕ35, то индицируется сообщение:

```
ОПРЕДЕЛИТЕ КОНФИГУРАЦИЮ СИСТЕМЫ
ПЗУ ЭМУЛЯТОРА ИЛИ ПЗУ МПУ
0 – ПЗУ ЭМУЛЯТОРА
1 – ПЗУ МПУ
```

Действия оператора: ввести 0 <ВК> или 1 <ВК>

Если набран другой символ, на экране появится сообщение: *ответ неверен, повторите!*

Следующие действия – выбор ОЗУ:

```
ОЗУ ДАННЫХ ЭМУЛЯТОРА ИЛИ МПУ
0 – ОЗУ ДАННЫХ ЭМУЛЯТОРА
1 – ОЗУ ДАННЫХ МПУ
```

Действия оператора: Ввести 0 <ВК> или 1 <ВК>

Если выбран режим эмуляции БИС КМ1816ВЕ48, то автоматически устанавливается память программ и память данных блока эмулятора. Что касается памяти программ, то работа блока эмулятора с памятью программ МПУ в этом режиме невозможна. Что касается памяти данных, то, вообще говоря, ОЭВМ в режиме работы с внутренней памятью может выполнить команду обращения к внешней памяти данных. Однако использование такой команды не рекомендовано, так как в результате портится содержимое порта Р0.

После выбора конфигурации отладочной системы следует выбор режима эмуляции:

РЕЖИМЫ ЭМУЛЯЦИИ
0 – НЕПРЕРЫВНЫЙ
1 – ПОШАГОВЫЙ
2 – С ОСТАНОВОМ ПО КТ

Действия оператора: Ввести 0 <ВК> или 1 <ВК>, или 2 <ВК>

В случае неправильного ввода последует сообщение: ***нет такого режима эмуляции***

Следующий запрос:

В КАКОЙ СИСТЕМЕ СЧИСЛЕНИЯ ВЫ ХОТИТЕ РАБОТАТЬ?
Н – 16-РИЧНАЯ, В – 8-РИЧНАЯ

Действия оператора: Ввести Н <ВК> или В<ВК>

Если введен непредусмотренный символ, последует сообщение: ***такая система счисления не предусмотрена***

Это сообщение будет повторяться до тех пор, пока не будет введен правильный символ.

Следующий запрос:

НА КАКОЕ УСТРОЙСТВО ХОТИТЕ ВЫВЕСТИ ИНФОРМАЦИЮ?
TT:, LPTT:

Действия оператора: Ввести TT: <ВК> если устройство отображения – дисплей.

Ввести LPTT: <ВК> если устройство отображения – одновременно дисплей и термопечатающее устройство.

Во втором случае, если не включена печать, следует сообщение: ***не включена печать***

Программа ожидает повторного ответа.

Следует иметь в виду, что если в качестве системного устройства используется накопитель на гибких магнитных мини-дисках типа 6022, то режим печати запрещен. Если в качестве системного устройства используется накопитель „Электроника ГМД 7012", то режим печати возможен. Это вызвано тем, что драйвер НГМД 6022 занимает достаточно большой объем ОЗУ БВК.

В конце начального диалога на экране дисплея появляется символ *, всегда означающий предложение пользователю ввести следующую команду.

Всего в распоряжении оператора имеется 18 команд. Ниже рассмотрены их назначение и особенности использования.

LOAD. По этой команде осуществляется загрузка отлаживаемой программы с накопителя ГМД или клавиатуры.

Пример. Загрузка с НГМД

```
* LOAD <ВК>
УКАЖИТЕ УСТРОЙСТВО, С КОТОРОГО БУДЕТ ВВОДИТЬСЯ ПРОГРАММА
DX: <ВК>
ИМЯ ФАЙЛА? PROBA. SAV <ВК>
АДРЕС ЗАГРУЗКИ = 400 В
ДЛИНА МОДУЛЯ = 5 В
ЗАГРУЗКА ПРОГРАММЫ ЗАКОНЧЕНА
*
```

Пример. Загрузка с клавиатуры.

```
* LOAD <ВК>
УКАЖИТЕ УСТРОЙСТВО, С КОТОРОГО БУДЕТ ВВОДИТЬСЯ ПРОГРАММА
TT: <ВК>
ЗАГРУЗКА ПРОГРАММЫ ПОЛЬЗОВАТЕЛЯ С ПУЛЬТА
ВВОД ЗАКОНЧЕН, ЕСЛИ КОМАНДА = 7777В ИЛИ FFFH
ВВЕДИТЕ НАЧАЛЬНЫЙ АДРЕС, А ЗАТЕМ ТОЛЬКО КОМАНДЫ – 0 <ВК>
0/47 <ВК>
1/27 <ВК>
2/67 <ВК>
3/4 <ВК>
4/0 <ВК>
5/7777 ВК
АДРЕС ЗАГРУЗКИ = 0 В
ДЛИНА ПРОГРАММЫ = 5 В
ЗАГРУЗКА ПРОГРАММЫ ЗАКОНЧЕНА
*
```

Если при выполнении команды запрос *имя файла* повторяется несколько раз, то это означает, что вызываемого файла нет на диске.

СНМЕМ. По команде СНМЕМ проводится изменение содержимого памяти программ эмулятора ROM, памяти данных эмулятора RAM, памяти данных контроллера CRAM, памяти данных МПУ USRAM. Оператор вводит адрес ячейки которую требуется изменить, и данные в выбранной системе счисления.

Пример.

```
* СНМЕМ <ВК>
КАКУЮ ПАМЯТЬ ВЫ ХОТИТЕ ИЗМЕНИТЬ?
ОТВЕТ МОЖЕТ БЫТЬ: ROM, RAM, CRAM, USRAM - RAM <ВК>
ВНЕСИТЕ ИЗМЕНЕНИЯ В ПАМЯТЬ ДАННЫХ ЭМУЛЯТОРА
ПРОЦЕСС ЗАКОНЧЕН, ЕСЛИ АДР > 377В ИЛИ FFFH
АДРЕС – 0 <ВК>
ИЗМЕНЕНИЯ – 47 <ВК>
АДРЕС – 400 <ВК>
ВВОД ИЗМЕНЕНИЙ ЗАКОНЧЕН
*
```

В случае, если оператор выбрал режим однокристальной микроЭВМ и указал адрес ПЗУ эмулятора более чем 1К, то индицируется сообщение: *несуществующий адрес.*

DEL. По команде DEL выполняется очистка памяти программ эмулятора.

Пример.

```
* DEL <ВК>
ОЧИСТКА ПАМЯТИ ПРОГРАММ ЭМУЛЯТОРА
ОЧИСТКА ЗАКОНЧЕНА
*
```

R. По команде R изменяется содержимое аккумулятора *A*, слова состояния процессора PSW, таймера T и портов ОЭВМ.

Пример.

```
* R <ВК>
КАКОЙ РЕГИСТР ВЫ ХОТИТЕ ИЗМЕНИТЬ? A <ВК>
ВНЕСИТЕ ИЗМЕНЕНИЯ В АККУМУЛЯТОР – 47 <ВК>
A = 47 B
*
```

Если при выполнении команды введено неправильное наименование регистра, то индицируется сообщение: *нет такого регистра*

TYPE. По команде TYPE выполняется вывод на индикацию содержимого памяти программ эмулятора ROM, памяти данных эмулятора RAM, памяти данных контроллера CRAM, памяти программ МПУ USROM, памяти данных МПУ USRAM.

Пример.

```
* TYPE <ВК>
КАКУЮ ПАМЯТЬ ВЫ ХОТИТЕ РАСПЕЧАТАТЬ?
ОТВЕТ МОЖЕТ БЫТЬ: ROM, PAM, USROM, USRAM, CRAM – ROM <ВК>
РАСПЕЧАТКА ПАМЯТИ ПРОГРАММ ЭМУЛЯТОРА
ВВЕДИТЕ НОМЕР СТРАНИЦЫ, КОТОРУЮ ВЫ ХОТИТЕ РАСПЕЧАТАТЬ –
0 <ВК>
```

В результате индицируется содержимое 256 ячеек указанной страницы памяти (память программ имеет страничную организацию). Индикация выполняется так же, как и для блока эмулятора БЭ-580. Если оператор указал номер несуществующей страницы памяти, то индицируется сообщение: *такой страницы нет.*

S. По команде S в случае останова индицируется мнемоника выполненной команды (при пошаговом режиме) и соедржимое доступных ресурсов ОЭВМ: счетчика команд, аккумулятора, слова состояния процессора с расшифровкой по признакам, таймера, портов P1 и P2, состояния выводов T0 и T1, а в режиме однокристальной микроЭВМ и порта P0 (шины BUS).

Пример.

```
* S <ВК>
PC = 1 CLR A
A=0 SP=0 RBO FO=0 AC=0 C=0 T=0 P1=377 B P2=377 B T0=0 T1=0
BUS=377 B
*
```

Если ранее был назначен непрерывный режим работы и программа выполняется, то индицируется сообщение: *идет выполнение программы.*

Отметим особенность индикации, которая имеет место в режиме ОЭВМ в случае, если пользователь ошибочно выведет данные на линии порта P0 со стороны МПУ и одновременно проведет запись данных в порт P0 из ОЭВМ. В этом случае на линиях порта P0 произойдет смешивание обоих данных. Для индикации состояния порта P0 (BUS) информация считывается в БВК непосредственно с линий порта, т. е. считываются смешанные данные. Если же выполняется команда INS A, BUS (ввод данных из порта P0 в A), то на время ее выполнения выходные линии порта переводятся в третье состояние и в A заносятся данные из МПУ без смешивания с данными порта P0. В результате после выполнения указанной команды содержимое A и порта P0 при индикации может различаться.

SEE. По команде SEE индицируется информация о выбранной конфигурации отладочной системы, системе счисления и режиме эмуляции.

Пример.

```
* SEE <ВК>
РЕЖИМ ОТЛАДОЧНОГО КРИСТАЛЛА:
ПЗУ МПУ И ОЗУ ЭМУЛЯТОРА
ВОСЬМЕРИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ
НЕПРЕРЫВНЫЙ РЕЖИМ
*
```

GO. Команда GO служит для запуска программы пользователя с необходимого адреса.

Пример.

```
* GO <ВК>
ВВЕДИТЕ АДРЕС ПУСКА ПРОГРАММЫ – 0 <ВК>
*
```

После команды GO рекомендуется подать команду S.

P. Команда P подается для продолжения выполнения программы при останове ОЭВМ. Процесс эмуляции продолжается с адреса останова. Индикация как при выполнении команды S.

ST. По команде ST в непрерывном режиме ОЭВМ выполняет шаг (одну команду) программы. Обычно команда ST выполняется после команды STOP. Индикация – как при выполнении команды S.

TRAC. Команда TRAC аналогична команде TPACE для блока эмулятора БЭ-580 (см. §3.1).

STOP. Команда STOP осуществляет останов ОЭВМ при работе в непрерывном режиме. Индикация - как при команде S, но без мнемоники выполненной команды.

RES. По команде RES на ОЭВМ подается управляющий сигнал по входу $\overline{\text{CLR}}$ (сброс).

NCP. По команде NCP осуществляется отмена контрольных точек по всему объему памяти программ.

Пример.

```
* NCP <ВК>
ОТМЕНА КТ
```

Далее следуют сообщения, как в команде REG.

CHANG. По команде CHANG осуществляется переназначение устройства, на которое будет выводиться индицируемая информация. Возможны два варианта: вывод на дисплей (TT:) или на дисплей и термопечать одновременно (LPTT:).

Пример.

```
* CHANG <ВК>
НА КАКОЕ УСТРОЙСТВО ХОТИТЕ ВЫВЕСТИ ИНФОРМАЦИЮ?
TT:, LPTT. – TT: <ВК>
```

CONF. По команде CONF осуществляется переопределение конфигурации системы. Команда выполняется аналогично назначению конфигурации в начальном диалоге.

REG. По команде REG изменяется режим эмуляции аналогично назначению режима эмуляции в начальном диалоге.

SYST. Команда SYST служит для изменения системы счисления и выполняется аналогично назначению системы счисления в начальном диалоге.

При работе с программой DEB816 оператору выдается ряд дополнительных сообщений об ошибках: НЕТ ТАКОЙ КОМАНДЫ

Если в названии команды была допущена ошибка, выдается сообщение: ОТВЕТ НЕВЕРЕН, ПОВТОРИТЬ

При невнимательной работе оператора, поступает сообщение:

В ДАННОЙ КОНФИГУРАЦИИ НЕТ ТАКОЙ ПАМЯТИ

Сообщение выдается при выполнении команды TYPE или CHMEM в случае, если неправильно введено предлагаемое название памяти или данная память не включена в конфигурацию отладочной системы.

Практические действия оператора по отладке МПУ аналогичны действиям при работе с программой DEB580 и определяются факторами, указанными в §3.1.

3.3. Отладка МПУ на базе микропроцессора КМ1810ВМ86

Состав АС для комплексной программно-аппаратурной отладки МПУ, построенного на базе МП КМ1810ВМ86, аналогичен комплексу для отладки МПУ на базе МП КР580ВМ80. Отличие заключается в использовании блока эмулятора БЭ-810 вместо БЭ-580.

Управляющая программа (монитор) комплекса MNTR86 для отладки устройств пользователя, построенных на базе МП КМ1810ВМ86, обеспечивает выполнение следующих функций: тестирование работы эмулятора и подключенных к нему устройств пользователя; загрузку программ пользователя с накопителя ГМД в память блока эмулятора или подключенного МПУ; запись отлаженных программ пользователя на ГМД; программную поддержку режимов отладки аппаратурной и программной компонент МПУ пользователя.

Управляющая программа представляет собой иерархию пяти мониторов, изображенную на рис. 3.1. Каждый из мониторов обеспечивает выполнение ряда однородных функций, необходимых для работы комплекса. При работе с любым монитором пользователь должен выбрать из перечня предлагаемых функций необходимую ему, нажать клавишу, код которой соответствует этой функции, и подтвердить свой выбор нажатием клавиши *ВК* или *ПС*. В процессе работы допускается коррекция введенных кодов с помощью клавиши *ЗБ*, которая отменяет последний введенный символ. Выполнение выбранной функции обеспечивается запуском соответствующей подпрограммы, причем возможно, что пользователю придется ответить на некоторые дополнительные вопросы, например ввести имя файла.

Такой подход к построению управляющей программы вызван желанием облегчить эксплуатацию комплекса пользователем, не являющимся профессиональным программистом. Процесс обучения работе на любом вычислительном комплексе легче, если он построен на диалоге обучаемого с ЭВМ. Для пользователей, освоивших приемы работы с данным комплексом, предусмотрен специальный режим, позволяющий опустить необходимость работы с большинством мониторов.

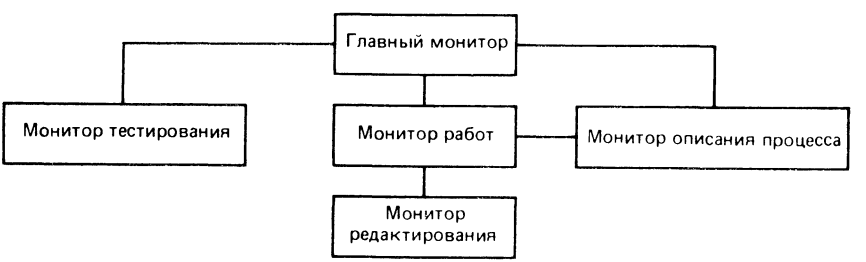


Рис. 3.1. Иерархия мониторов управляющей программы MNTR86

Главный монитор. При запуске отладочного комплекса осуществляется автоматическое тестирование связи БВК с блоком эмулятора, ОЗУ эмулятора и ЦП блока эмулятора. После этого в ОЗУ эмулятора загружается служебная программа, и на экране дисплея БВК выдается меню главного монитора:

Код	Функция
A	Чтение файла
B	Запись файла
C	Чтение файла состояния
D	Запись файла состояния
E	Сброс эмулятора
F	Вызов монитора работ
G	Вызов монитора описания процесса
H	Вызов монитора тестирования

Индикация меню свидетельствует о готовности комплекса к работе. При обнаружении сбоев в процессе тестирования пользователю выдаются соответствующие сообщения.

Используя главный монитор, пользователь может осуществить начальный сброс блока эмулятора, считать или записать файл на НГМД или вызвать другой монитор. Большинство функций являются очевидными, за исключением функций чтения и записи файла состояния. Под *файлом состояния* понимается файл, содержащий значение всех регистров МП на некоторый определенный момент времени. Если по какой-то причине пользователю необходимо прервать свою работу с комплексом, но отладка еще не закончена, то, записав файл состояния, можно продолжить работу после перерыва точно из того состояния, в котором пользователь остановился.

Монитор тестирования. Меню монитора тестирования представлено ниже:

Код	Функция
A	Тест ОЗУ эмулятора
B	Тест ОЗУ устройства пользователя
C	Цикл чтения ячейки памяти
D	Цикл записи ячейки памяти
E	Цикл чтения и записи ячейки памяти
F	Цикл чтения порта
G	Цикл записи порта
H	Вызов главного монитора

Этот монитор призван обеспечить проверку работоспособности блока эмулятора, а также облегчить наладку МПУ пользователя. Для этого в ОЗУ эмулятора загружается, а затем выполняется соответствующая тестирующая программа. Выполнение всех тестов осуществляется ЦП блока эмулятора. Проверка ОЗУ эмулятора выполняется БВК. Перед выполнением любого теста, кроме теста ОЗУ эмулятора, у пользователя запрашивается дополнительная информация, необходимая для работы тестов: адрес ячейки памяти, номер порта и т. п.

Монитор описания процесса. Меню монитора описания процесса представлено ниже:

Код	Функция
A	Задание точек останова
B	Задание стартового адреса
C	Перемещение монитора эмулятора
D	Листинг процесса
E	Вызов монитора работ

Этот монитор служит для задания условий протекания процесса отладки.

Задание точек останова позволяет пользователю прерывать процесс эмуляции в интересующих его точках. Комплекс поддерживает четыре возможных условия прерывания эмуляции: по адресу, по записи, по значению, трассировка в заданном интервале. Останов по адресу вызывает прерывание эмуляции, когда значение регистра адреса команд становится равным заданному пользователем. Останов по записи вызывает прерывание эмуляции в случае изменения содержимого заданной ячейки ОЗУ. Останов по значению прерывает эмуляцию в случае, если содержимое заданной ячейки ОЗУ становится равным заданному значению. Задание интервала трассировки служит для обеспечения трассирования заданного пользователем фрагмента программы.

Задание стартового адреса обеспечивает запуск программы пользователя с определенного адреса независимо от текущего состояния регистров МП.

Функция перемещения монитора эмулятора позволяет записать служебную программу блока эмулятора в заданную пользователем область ОЗУ эмулятора.

Функция листинга процесса обеспечивает индикацию на экране дисплея БВК всех точек останова, стартовый адрес и начальный адрес загрузки служебной программы.

Для удобства пользователя из монитора описания процесса предусмотрен вызов монитора работ.

Монитор редактирования. Меню монитора редактирования представлено ниже:

Код	Функция
A	Коррекция регистров
B	Коррекция памяти
C	Просмотр памяти
D	Шестнадцатеричная арифметика
E	Перемещение массива
F	Вызов монитора работ

С помощью этого монитора пользователь может скорректировать содержимое регистров МП, просмотреть и модифицировать ОЗУ эмулятора или МПУ пользователя. Сервисной функцией монитора является возможность для пользователя складывать и вычитать числа в шестнадцатерич-

ном коде. Коррекция ОЗУ выполняется в трех режимах: коррекция одного байта, заполнение массива списком байт и заполнение массива однородной информацией.

Монитор работ. Меню монитора работ представлено ниже:

Код	Функция	Режим
A	Вызов главного монитора	
B	Вызов монитора описания процесса	
C	Вызов монитора редактирования	
D	Сброс	
E	RUN	
F	CONT	
G	STEP	
H	STOP	
I	DISP/PRINT	DISP
J	CHECK/NON	NON
K	TPACE/NORMAL	NORMAL
L	EXPRESS – DEBUGGER	

Основными функциями этого монитора являются запуск и останов программы пользователя, а также задание некоторых специальных режимов.

Программу пользователя запускают в соответствии с одной из трех функций: RUN, CONT, STEP. Причем функция CONT обеспечивает запуск программы с текущего адреса с сохранением значений всех регистров, т. е. продолжение программы, а функция RUN вызывает запуск программы со стартового адреса, причем все регистры обнуляются, за исключением стековых.

Программа может выполняться в следующих режимах: с контролем и без контроля точек останова, с трассировкой или без трассировки выполняемой программы, с выводом затребованной пользователем информации на экран дисплея или на печатающее устройство.

Последнюю функцию монитора работ, вызываемую нажатием клавиши L, называют быстрый отладчик (EXPRESS – DEBUGGER). Эта функция позволяет опытному пользователю избегать необходимости обязательного прохода по всем уровням иерархии мониторов при эксплуатации комплекса. При входе в эту функцию меню монитора перестают выдаваться на экран дисплея, а выполнение требуемых для работы команд:

Код	Команда	Входные параметры	Примечания
I	INPUT	<АДРЕС>:<БАЙТ>	Ввод одного байта
I	INPUT	<АДРЕС>:<БАЙТ>: <БАЙТ>:<БАЙТ>: . . .	Ввод списка байт
I	INPUT	<АДРЕС>:<АДРЕС>: <СЛОВО>	Заполнение массива заданным словом
L	LOOK	<АДРЕС>	Просмотр слова
L	LOOK	<АДРЕС>:<АДРЕС>	Просмотр массива
M	MOVE	<АДРЕС>:<АДРЕС>: <АДРЕС>	Перемещение массива

H	HEX	<ЧИСЛО>:<ЧИСЛО>	Сложение и вычитание пары чисел в шестнадцатеричном формате
G	GO	<АДРЕС> – необязательный параметр	Запуск программы с текущего или заданного адреса
S	STEP	Нет	Шаг программы
R	REGISTER	Нет	Коррекция регистров
P	POINT	Нет	Задание точек останова

осуществляется нажатием специальных управляющих клавиш, причем для некоторых команд необходимо задать входные параметры.

Команда INPUT предназначена для ввода информации в ОЗУ эмулятора или МПУ: для ввода одного байта указываются адрес и содержимое байта, для ввода списка байт указываются начальный адрес и список байт, для заполнения массива заданным словом указываются начальный и конечный адреса массива и требуемое слово. Завершается ввод команды нажатием клавиши *I* (код команды).

Команда LOOK позволяет вывести на экран дисплея содержимое памяти блока эмулятора или МПУ: для вывода слова указывается его адрес, а для вывода массива – его начальный и конечный адреса. Завершается ввод команды нажатием клавиши *L*. Как указывалось в разделе 2.5, адрес ячейки памяти определяет ее принадлежность к памяти эмулятора или МПУ.

Команда MOVE позволяет переместить массив в необходимую область памяти. При этом указываются начальный и конечный адреса перемещаемого массива и адрес его нового местоположения.

Команда HEX проводит сложение и вычитание двух чисел в шестнадцатеричном коде. После выполнения команды на экран дисплея выводятся их сумма и разность.

Команда GO осуществляет запуск МП на выполнение программы с заданного адреса. По умолчанию запуск выполняется с текущего адреса.

Команда STEP позволяет выполнить шаг программы.

Команда REGISTER осуществляет вывод на дисплей содержимого всех программно-доступных регистров МП, при необходимости возможна их коррекция.

Команда POINT позволяет задать точки основа. Условия останова задаются в ходе диалога с оператором.

```
PROGRAM DEB580;
LABEL 999;
CONST MEMORY=100000B;
      GENER=40000B;
      CR=20000B;
      ZAPNOMEM=4000B;
      HUND=1000B;
      SR=10000B;
TYPE  INT=INTEGER;
      LODE=ARRAY [1..2] OF CHAR;
VAR   RAB:INT;
      OUT:LODE;
      FTXT:TEXT;
      KOD,KOD1,KOD2:INT;
      C1,C2,C3,C4: CHAR;
      AME ORIGIN 165000B,
      DME ORIGIN 165002B,
      DMU ORIGIN 165004B,
      RS ORIGIN 165006B:INT;
      ANS:0..2;
      NAME: ARRAY [1..15] OF CHAR;
      MAX,SDV,DMOD,AZAG: INT;
      DATA,NI,NJ,NK:INT;
      ASKL,SKL1,SKL2,SKL3:INT;
      US,SS,K,M:INT;
      COM:ARRAY [1..4] OF CHAR;
      D,RIJ,ADR:INT;
      FL,A,B,C,D,BE:INT;
      FC,FP,FCY,FZ,FS:0..1;
      E,H,L,SP:INT;
      N0,N1,PC,PC0:INT;
      IM:ARRAY[1..2] OF CHAR;
      VAL,P0,P1,P2:INT;
      REG,SHAG,WAIT,Z16,SYST:BOOLEAN;
      COM1,COM2,COM3:INT;
/*================================*/
PROCEDURE RD;EXTERNAL;
PROCEDURE DISAS;EXTERNAL;
/*================================*/
PROCEDURE NEPREG(VAR K,AZAG:INT);
VAR I,J:INT;
BEGIN I:=K;J:=AZAG;/*ВС
NEP0:   MOV J(6),R0
        MOV I(6),R1
        BIC #^0177400,R1
        MOV R0,AME
        MOV R1,DME
        MOV DME,R3
        BIC #^0177400,R3
        CMP R1,R3
        BEQ NEP1
        BR NEP0
NEP1:   NOP
*/;END;
/*================================*/
PROCEDURE OPROS;
LABEL 11;
VAR I:INT;
BEGIN FOR I:=1 TO 512 DO;/*ВС
        CLR I(6)
        MOV #10,R0
OP1:    BIT #^01000,DME
        BNE OP2
        INC I(6)
```

```
OP2:    SOB R0,OP1
*/;IF (DME AND MEMORY)=0 THEN
 BEGIN IF I<>0 THEN
        BEGIN
     WRITELN('ВЫПОЛНЕНИЕ ПРОГРАММЫ...');
 GOTO 999;END ELSE GOTO 11;END ELSE
WRITELN('ЦП В ЗАХВАТЕ.');GOTO 999;
11: WAIT:=TRUE;RS:=5;
END;
/*=========================*/
PROCEDURE OCHMEM;
BEGIN WRITE('ОЧИСТКА ПАМЯТИ ЭМУЛЯТОРА- ');/*ВС
        MOV MAX(5),R0
        CLR R1
CYKL:   MOV R1,AME
        MOV #0,DME
        MOV DME,R2
        BIC #^O177000,R2
        CMP #0,R2
        BNE CYKL
        INC R1
        SOB R0,CYKL
        MOV #^O10000,R0
        MOV SDV(5),R1
OCH1:   MOV R1,AME
        MOV #0,DME
        MOV DME,R2
        BIC #^O177000,R2
        CMP #0,R2
        BNE OCH1
        INC R1
        SOB R0,OCH1
*/;WRITELN('-ОЧИСТКА ЗАКОНЧЕНА');END;
/*=========================*/
PROCEDURE INP8;
LABEL 1,2;
VAR X1,X2,X3,X4,X5,X6:INT;
AD:ARRAY [1..6] OF CHAR;
X:INT;
PROCEDURE RAZR(AD: CHAR);
BEGIN X:=ORD(AD)-60B;
IF X=-16 THEN BEGIN X:=0;BIJ:=BIJ+1;END;
IF X IN [0..7] THEN X:=X ELSE BEGIN WRITE('НЕ ВОСЬМЕРИЧНОЕ ЧИСЛО!'
WRITELN('ПОВТОРИТЕ ВВОД !');WRITE('?');GOTO 1;END;END;
BEGIN 1:READLN(AD);BIJ:=0;
RAZR(AD[1]);X1:=X;
RAZR(AD[2]);X2:=X;
RAZR(AD[3]);X3:=X;
RAZR(AD[4]);X4:=X;
RAZR(AD[5]);X5:=X;
RAZR(AD[6]);X6:=X;
CASE BIJ OF
0: GOTO 2;
1: BEGIN X6:=X5;X5:=X4;X4:=X3;X3:=X2;X2:=X1;X1:=0 END;
2: BEGIN X6:=X4;X5:=X3;X4:=X2;X3:=X1;X1:=0;X2: 0 END;
3: BEGIN X6:=X3;X5:=X2;X4:=X1;X1:=0;X2:=0;X3:=0 END;
4: BEGIN X6:=X2;X5:=X1;X1:=0;X2:=0;X3:=0;X4:=0 END;
5: BEGIN X6:=X1;X1:=0;X2:=0;X3:=0;X4:=0;X5:=0 END;
6: BEGIN WRITE('ВВЕДИТЕ ЧИСЛО! -');GOTO 1;END;
END;
2: ADR:=X6+X5*10B+X4*100B+X3*1000B+X2*10000B+X1*100000B;
END;
/*=============================*/
PROCEDURE INP16;
```

```
LABEL 1,2;
VAR X,X1,X2,X3,X4:INTEGER;
    AD:ARRAY [1..4] OF CHAR;
PROCEDURE SMENA(AD: CHAR);
BEGIN   X:=ORD(AD)-60B;
        IF X=-16 THEN BEGIN BIJ:=BIJ+1;X:=0 END;
        IF X IN [0..9,17..22] THEN
                CASE X OF
                0: X:=0;
                1: X:=1;
                2: X:=2;
                3: X:=3;
                4: X:=4;
                5: X:=5;
                6: X:=6;
                7: X:=7;
                8: X:=8;
                9: X:=9;
               17: X:=10;
               18: X:=11;
               19: X:=12;
               20: X:=13;
               21: X:=14;
               22: X:=15;
                END
        ELSE BEGIN WRITE ('НЕ ШЕСТНАДЦАТЕРИЧНОЕ ЧИСЛО!');
        WRITELN('ПОВТОРИТЕ ВВОД ');WRITE('?');GOTO 1;END;END;
BEGIN 1:READLN(AD);BIJ:=0;
        SMENA(AD[1]);X1:=X;
        SMENA(AD[2]);X2:=X;
        SMENA(AD[3]);X3:=X;
        SMENA(AD[4]);X4:=X;
        CASE BIJ OF
        0: GOTO 2;
        1: BEGIN X4:=X3;X3:=X2;X2:=X1;X1:=0 END;
        2: BEGIN X4:=X2;X3:=X1;X1:=0;X2:=0 END;
        3: BEGIN X4:=X1;X1:=0;X2:=0;X3:=0 END;
        4: BEGIN WRITE('ВВЕДИТЕ ЧИСЛО!-');GOTO 1;END;
        END;
      2:BIJ:=X;
        /*BC
        MOV X1(6),R1
        SWAB R1
        ASL R1
        ASL R1
        ASL R1
        ASL R1
        MOV X2(6),R0
        SWAB R0
        ADD R0,R1
        MOV X3(6),R0
        ASL R0
        ASL R0
        ASL R0
        ASL R0
        ADD R0,R1
        ADD X4(6),R1
        MOV R1,ADR(5)
        */;END;
/*============================*/
PROCEDURE OUT16;
LABEL 1;
VAR TR: BOOLEAN;
    X,X1,X2,X3,X4:INT;
```

```
    C:CHAR;
PROCEDURE ZNAK(X:INT);
BEGIN CASE X OF
        0B: C:='0';
        1B: C:='1';
        2B: C:='2';
        3B: C:='3';
        4B: C:='4';
        5B: C:='5';
        6B: C:='6';
        7B: C:='7';
       10B: C:='8';
       11B: C:='9';
       12B: C:='A';
       13B: C:='B';
       14B: C:='C';
       15B: C:='D';
       16B: C:='E';
       17B: C:='F';
        END;END;
BEGIN   /*BC
        MOV DATA(5),R0
        MOV R0,X1(6)
        BIC #^O177760,X1(6)
        ASR R0
        ASR R0
        ASR R0
        ASR R0
        MOV R0,X2(6)
        BIC #^O177760,X2(6)
        ASR R0
        ASR R0
        ASR R0
        ASR R0
        MOV R0,X3(6)
        BIC #^O177760,X3(6)
        ASR R0
        ASR R0
        ASR R0
        ASR R0
        MOV R0,X4(6)
        BIC #^O177760,X4(6)
        */;ZNAK(X1);C1:=C;
        ZNAK(X2);C2:=C;
        ZNAK(X3);C3:=C;
        ZNAK(X4);C4:=C;IF Z16 THEN GOTO 1;
        TR:=(C4='0') AND (C3='0') AND (C2='0');
        IF TR=TRUE THEN BEGIN WRITE(C1); GOTO 1; END
        ELSE TR:=(C4='0') AND (C3='0');
        IF TR=TRUE THEN BEGIN WRITE(C2,C1);GOTO 1; END
        ELSE TR:=(C4='0');
        IF TR=TRUE THEN BEGIN WRITE(C3,C2,C1);GOTO 1;END
        ELSE WRITE(C4,C3,C2,C1);
1: END;
/*==============================*/
PROCEDURE I816;
BEGIN IF SYST THEN INP8 ELSE INP16;END;
/*==============================*/
PROCEDURE CONFMEM;
BEGIN
WRITE('ЗАДАЙТЕ НАЧАЛЬНЫЙ АДРЕС ПЕРЕМЕЩАЕМОГО БЛОКА ОЗУ ЭМУЛЯТОРА-');
I816;SDV:=ADR;RS:=SDV OR 1;
END;
/*==========================*/
```

```
PROCEDURE ZAĎVOL;
BEGIN MAX:=0;/*ВС
VOLO:    MOV MAX(5),AME
         MOV #^O123,DME
         MOV DME,R0
         BIC #^O177400,R0
         CMP #^O377,R0
         REQ VOL1
         ADD #^O10000,MAX(5)
         BR VOLO
VOL1:    NOP
*/;IF MAX=0 THEN
WRITELN('ПАМЯТЬ ОТКЛЮЧЕНА''') ELSE BEGIN
WRITE('ОБ''ЕМ ПАМЯТИ ЭМУЛЯТОРА:0-');
IF SYST THEN WRITELN(MAX-1:-1) ELSE
BEGIN DATA:=MAX-1;OUT16;WRITELN;END;
END;END;
/*=========================*/
PROCEDURE MNEM;
LABEL 1;
VAR RPC,I:INT;
BEGIN OPROS;US:=RS AND 177000B;RS:=5;0:=AME;AME:=0;RS:=US OR 400B;
PC:=AME;AME:=PC0;
/*ВС
         TST US(5)
         BMI MN1
         MOV DME,KOD(5)
         BR MNO
MN1:     MOV DMU,KOD(5)
MNO:     INC AME
         TST US(5)
         BMI MN3
         MOV DME,KOD1(5)
         BR MN2
MN3:     MOV DMU,KOD1(5)
MN2:     INC AME
         TST US(5)
         BMI MN5
         MOV DME,KOD2(5)
         BR MN4
MN5:     MOV DMU,KOD2(5)
MN4:     MOVB KOD1(5),I(6)
         MOVB KOD2(5),I+1(6)
         BIC #^O177400,KOD(5)
         BIC #^O177400,KOD1(5)
         BIC #^O177400,KOD2(5)
*/;IF COM3=43B THEN GOTO 1;IF (RS AND MEMORY)=100000B THEN
IF (RS AND 377B)=43B THEN
WRITE('ВЫПОЛНЕНИЕ КОМАНДЫ: ') ELSE WRITE('ВЫПОЛНЕНА КОМАНДА: ') ELSE
WRITE('ВЫПОЛНЕНА КОМАНДА: ');
IF ((KOD=300B) OR (KOD=307B) OR (KOD=310B) OR (KOD=311B) OR (KOD=166B)
OR (KOD=317B) OR (KOD=320B) OR (KOD=327B) OR (KOD=330B) OR (KOD=337B)
OR (KOD=340B) OR (KOD=347B) OR (KOD=350B) OR (KOD=357B) OR (KOD=360B)
OR (KOD=367B) OR (KOD=351B) OR (KOD=370B) OR (KOD=377B)) THEN
 BEGIN DISAS;WRITELN;GOTO 1;END;
RPC:=PC-PC0;
IF RPC=1 THEN BEGIN DISAS;WRITELN;END;
IF RPC=2 THEN BEGIN DISAS;IF SYST THEN
   WRITELN(KOD1:-1) ELSE BEGIN DATA:=KOD1;OUT16;WRITELN;END;END;
IF ((RPC>=3) OR (RPC<1)) THEN BEGIN
   DISAS;IF SYST THEN WRITELN(I:-1) ELSE BEGIN DATA:=I;OUT16;WRITELN;
END; END; 1: 0:=AME;AME:=0;RS:=US AND 177000B;
END;
/*=========================*/
```

```
PROCEDURE TYPEMEM;
PROCEDURE PUNCH;
VAR I,K,J,S,L,X:INT;
BEGIN IF SYST THEN BEGIN
WRITE('  АДРЕС');
WRITELN('0':3,'1':6,'2':6,'3':6,'4':6,'5':6,'6':6,'7':6);END
ELSE BEGIN
WRITE('АДРЕС');
WRITE('0':3,'1':4,'2':4,'3':4,'4':4,'5':4,'6':4,'7':4);
WRITELN('8':4,'9':4,'A':4,'B':4,'C':4,'D':4,'E':4,'F':4);END;
WRITELN;
I:=ADR;Z16:=TRUE;
IF SYST THEN BEGIN S:=16;
FOR J:=0 TO S DO
BEGIN WRITE(I:-6,'/');
FOR L:=0 TO 7 DO
BEGIN /*ВС
        MOV I(6),AME
        TST US(5)
        BMI TY0
        MOV DME,K(6)
        BR TY1
TY0:    BIC #^02000,SS(5)
        MOV SS(5),RS
        MOV DME,KOD(5)
        BIC #^0177377,KOD(5)
        MOV DMU,K(6)
        BIC #^0177400,K(6)
        BIS KOD(5),K(6)
TY1:    BIC #^0177000,K(6)
        MOV K(6),R0
        ASR R0
        ASR R0
        ASR R0
        ASR R0
        ASR R0
        ASR R0
        MOV R0,X(6)
*/ IF X=0 THEN WRITE(' ',K:-3,'  ')
ELSE WRITE(K:-3,'  ');I:=I+1;K:=K+1 END;WRITELN;END;END ELSE
BEGIN S:=17;FOR J:=0 TO S DO
BEGIN DATA:=I;OUT16;WRITE(C4,C3,C2,C1,'/');
FOR L:=0 TO 15 DO
BEGIN /*ВС
        MOV I(6),AME
        TST US(5)
        BMI TY2
        MOV DME,K(6)
        BR TY3
TY2:    BIC #^02000,SS(5)
        MOV SS(5),RS
        MOV DME,KOD(5)
        BIC #^0177377,KOD(5)
        MOV DMU,K(6)
        BIC #^0177400,K(6)
        BIS KOD(5),K(6)
TY3:    BIC #^0177000,K(6)
*/;DATA:=K;OUT16;IF C3='0' THEN C3:=' ' ELSE C3:='*';
WRITE(C3,C2,C1,' ');I:=I+1;K:=K+1;END;WRITELN;END;END;
END;
BEGIN IF(DME AND HUND)=0 THEN
BEGIN IF(DME AND MEMORY)=0 THEN
WRITELN('ВЫПОЛНЕНИЕ ПРОГРАММЫ') ELSE
WRITELN('ЦП В ЗАХВАТЕ.');
```

```
GOTO 999;END ELSE
BEGIN US:=RS AND 177400B;RS:=5;0:=AME;AME:=0;PC:=AMF;
RS:=US OR 6400B;SS:=RS AND 177400B;
WRITE('НАЧАЛЬНЫЙ АДРЕС?');I816;
WRITE('                СОДЕРЖИМОЕ ПАМЯТИ ');
IF (US AND MEMORY)=0 THEN WRITELN('ЭМУЛЯТОРА') ELSE
WRITELN('ПОЛЬЗОВАТЕЛЯ');PUNCH;Z16:=FALSE;
END;AME:=PC;RS:=US;
END;
/*============================*/
PROCEDURE CIREG;
VAR I:INT;
BEGIN WRITELN(' ':20,'ЦИКЛИЧЕСКИЙ РЕЖИМ');
WRITE('АДРЕС ПЕРВОЙ ВЫПОЛНЯЕМОЙ КОМАНДЫ-?');I816;
I:=ADR;US:=RS AND 177000B;0:=AME;AME:=0;RS:=US OR 400B;
WRITE('АДРЕС ПОСЛЕДНЕГО БАЙТА ЗАЦИКЛИВАЕМОГО УЧАСТКА-?');
I816;ASKL:=ADR+1;/*ВС
        MOV ASKL(5),AME
        MOV DME,SKL1(5)
        MOV #^0303,DME
        INC AME
        MOV DME,SKL2(5)
        MOV #0,DME
        MOVB I(6),DME
        INC AME
        MOV DME,SKL3(5)
        MOV #0,DME
        MOVB I+1(6),DME
*/;AME:=0;RS:=US;
WRITELN('ДАЙТЕ КОМАНДУ НА ЗАПУСК С НАЧАЛЬНОГО АДРЕСА');
GOTO 999;END;
/*============================*/
PROCEDURE SPECPR;
BEGIN /*ВС
        MOV #^07720,R0
        MOV #31,R1
        MOV #SPEC1,R2
SPEC0:  MOV R0,AME
        MOV (R2)+,DME
        INC R0
        SOB R1,SPEC0
        BR SPEC2
SPEC1:  .WORD ^042,^0305,^017,^041,0,0,^0332,^0335,^017
        .WORD ^071,^0303,^0337,^017,^071,^067
        .WORD ^061,^0317,^017,^0345,^0365,^0305
        .WORD ^0325,^0321,^0301,^0361,^0341,^0371
        .WORD ^052,^0305,^017,^0303
SPEC2:  MOV R0,AME
        MOVB PC(5),DME
        INC AME
        MOVB PC+1(5),DME
*/ END;
/*============================*/
PROCEDURE STORE;
BEGIN /*ВС
        MOV #^07705,R0
        MOV #44,R1
        MOV #ST1,R2
STO:    MOV R0,AME
        MOV DME,(R2)+.
        INC R0
        SOB R1,STO
        BR ST2
ST1:    .WORD 0,0,0,0,0,0,0,0,0,0,0,0
```

```
        .WORD 0,0,0,0,0,0,0,0,0,0,0
        .WORD 0,0,0,0,0,0,0,0,0,0,0
        .WORD 0,0,0,0,0,0,0,0,0,0,0
ST2:    NOP
*/;END;
/*=========================*/
PROCEDURE RESTORE;
BEGIN /*BC
        MOV #^O7705,R0
        MOV #44,R1
        MOV #ST1,R2
RES:    MOV R0,AME
        MOV (R2)+,DME
        INC R0
        SOB R1,RES
*/ END;
/*============================*/
PROCEDURE JUMP;
LABEL 1;
VAR I,A,B:INT;
BEGIN PC:=AME;A:=DME;DME:=303B;
 AME:=AME+1;B:=DME;DME:=320B;AME:=AME+1;COM3:=DME;DME:=17B;
 GE:=DME AND 377B;IF GE=17B THEN BEGIN AME:=0;RS:=SS AND 177000B;
 RS:=6;
1:  I:=RS AND HUND;IF I=0 THEN GOTO 1;0:=AME;AME:=0;
 RS:=SS OR 400B;END;AME:=PC;DME:=A;AME:=AME+1;DME:=B;AME:=AME+1;
 DME:=COM3;END;
/*============================*/
PROCEDURE UKPECH;
BEGIN WRITELN('ВЫВОД СОДЕРЖИМОГО РОН НЕВОЗМОЖЕН.');
COM3:=RS AND 377B;
WRITE('ШИНА АДРЕСА=');
IF SYST THEN WRITE(COM1:-1) ELSE BEGIN DATA:=COM1;OUT16;END;
WRITE(' ':3);WRITE('ШИНА ДАННЫХ=');
IF SYST THEN WRITE(COM2:-1) ELSE BEGIN DATA:=COM2;OUT16;END;
WRITE(' ':3);WRITE('ДЕЙСТВИЕ ЦП: ');
IF COM3=242B THEN WRITELN('ВЫБОРКА КОДА ОПЕРАЦИИ');
IF COM3=202B THEN WRITELN('СЧИТЫВАНИЕ ПАМЯТИ');
IF COM3=0 THEN WRITELN('ЗАПИСЬ В ПАМЯТЬ');
IF COM3=206B THEN WRITELN('СЧИТЫВАНИЕ СТЕКА');
IF COM3=4 THEN WRITELN('ЗАПИСЬ В СТЕК');
IF COM3=102B THEN WRITELN('ВВОД ОТ ВНЕШНЕГО УСТРОЙСТВА');
IF COM3=20B THEN WRITELN('ВЫВОД НА ВНЕШНЕЕ УСТРОЙСТВО');
IF COM3=43B THEN WRITELN('ОБРАБОТКА ПРЕРЫВАНИЯ');
IF COM3=212B THEN WRITELN('ПОДТВЕРЖДЕНИЕ ОСТАНОВА');
IF COM3=53B THEN WRITELN('ПОДТВЕРЖДЕНИЕ ПРЕРЫВАНИЯ ПРИ ОСТАНОВЕ');
END;
/*=========================*/
PROCEDURE INDINF;
LABEL 1,2;
VAR I:INT;
BEGIN RS:=5;US:=RS AND 177000B;0:=AME;AME:=0;
COM1:=AME;COM2:=DME AND 377B;RS:=US OR 6400B;
I:=RS AND 51B;IF I<>40B THEN UKPECH ELSE BEGIN SS:=RS AND 57400B;
SS:=SS OR 16000B;RS:=SS;JUMP;IF GE<>17B THEN BEGIN UKPECH;GOTO 2;
END;STORE;SPECPR;AME:=0;RS:=SS AND 177000B;FOR I:=1 TO 17 DO RS:=6;
1: I:=RS AND HUND;IF I=0 THEN GOTO 1;0:=AME;AME:=0;
RS:=SS OR 400B;/*BC
        MOV #^O7705,AME
        MOVB DME,L(5)
        MOV #^O7706,AME
        MOVB DME,H(5)
        MOV #^O7707,AME
        MOVB DME,E(5)
```

```
        MOV #^O7710,AME
        MOVB DME,D(5)
        MOV #^O7711,AME
        MOVB DME,C(5)
        MOV #^O7712,AME
        MOVB DME,B(5)
        MOV #^O7713,AME
        MOVB DME,FL(5)
        MOV #^O7714,AME
        MOVB DME,A(5)
        MOV #^O7715,AME
        MOVB DME,SP(5)
        MOV #^O7716,AME
        MOVB DME,SP+1(5)
        BIC #^O177400,A(5)
        BIC #^O177400,B(5)
        BIC #^O177400,C(5)
        BIC #^O177400,D(5)
        BIC #^O177400,H(5)
        BIC #^O177400,L(5)
        BIC #^O177400,E(5)
        BIC #^O177400,FL(5)
#/;RESTORE; IF SYST THEN BEGIN
WRITE('B=<',B:-3,'> C=<',C:-3,'> D=<',D:-3,'> E=<',E:-3,'> H=< ,H:-3);
WRITELN('> L=<',L:-3,'>');
WRITE('A=<',A:-3,'> SP=<',SP:-6,'> PC=<');
WRITELN(PC:-6,'>');END ELSE
BEGIN WRITE('A=<');DATA:=A;OUT16;
WRITE('> B=<');DATA:=B;OUT16;WRITE('> C=<');DATA:=C;OUT16;
WRITE('> D=<');DATA:=D;OUT16;WRITE('> E=<');DATA:=E;OUT16;
WRITE('> H=<');DATA:=H;OUT16;WRITE('> L=<');DATA:=L;OUT16;
WRITE('> SP=<');DATA:=SP;OUT16;
WRITE('> PC=<');
DATA:=PC;OUT16;WRITELN('>');END;
IF (FL AND 1B)=0 THEN FC:=0 ELSE FC:=1;
IF (FL AND 4B)=0 THEN FP:=0 ELSE FP:=1;
IF (FL AND 20B)=0 THEN FCY:=0 ELSE FCY:=1;
IF (FL AND 100B)=0 THEN FZ:=0 ELSE FZ:=1;
IF (FL AND 200B)=0 THEN FS:=0 ELSE FS:=1;
WRITE('БИТЫ УСЛОВИЙ ЦП:');
WRITELN(' C=',FC:1,' P=',FP:1,' AC=',FCY:1,' Z=',FZ:1,' S=',FS:1);END;
2: AME:=Q;RS:=US;END;
/*=========================*/
PROCEDURE RES;
LABEL 1,2;
VAR I:INT;
BEGIN RS:=5;US:=RS AND 177000B;RS:=US OR OR;RS:=4;
SS:=US AND 57000B;RS:=SS;SS:=SS OR 16000B;RS:=SS;SS:=SS OR 400B;
Q:=AME;AME:=Q;RS:=SS;JUMP;
STORE;SPECPR;AME:=Q;RS:=SS AND 177000B;
FOR I:=1 TO 10 DO RS:=6;
1: I:=RS AND HUND;IF I=0 THEN GOTO 1;Q:=AME;AME:=0;RS:=SS OR 400B;/*PC
        MOV #10,R1
        MOV #^O7705,AME
RESV:   MOV #0,DME
        INC AME
        SOB R1,RESV
*/;AME:=0;RS:=SS AND 177000B;
FOR I:=1 TO 7 DO RS:=6;2: I:=RS AND HUND;IF I=0 THEN GOTO 2;
Q:=AME;AME:=0;RS:=SS OR 400B;RESTORE;AME:=Q;RS:=US;
END;
/*=========================*/
PROCEDURE IZMMEM;
LABEL 5,51;
```

```
VAR I:INT;
BEGIN US:=RS AND 177400B;RS:=5;0:=AME;AME:=0;RS:=US OR 6000B;
RS:=US OR 6400B;
WRITE('АДРЕС ЯЧЕЙКИ ПАМЯТИ-');I816;/*ВС
        MOV ADR(5),AME
        MOV DME,I(6)
        BIC #^O177000,I(6)
*/;IF SYST THEN WRITELN(ADR:-6,'=<',I:-3,'>') ELSE BEGIN
DATA:=ADR;OUT16;WRITE('=<');DATA:=I;OUT16;WRITELN('>');END;
WRITE('ХОТИТЕ ИЗМЕНИТЬ СОДЕРЖИМОЕ ПАМЯТИ');
5: WRITE('(ОТВЕТ ДОЛЖЕН БЫТЬ:0-НЕТ,1-ДА)?');READLN(ANS);
IF ANS=1 THEN BEGIN
WRITE('ВВЕДИТЕ НОВОЕ СОДЕРЖИМОЕ-');I816;/*ВС
        MOV ADR(5),DME
*/;END ELSE
IF ANS=0 THEN GOTO 51 ELSE
BEGIN WRITELN('ОТВЕТ НЕВЕРЕН,ПОВТОРИТЕ!');GOTO 5;END;
51: WRITELN;AME:=0;RS:=US;
END;
/*========================*/
PROCEDURE MODREG;
VAR V:INT;
BEGIN
WRITE('ИМЯ РОН,СОДЕРЖИМОЕ КОТОРОГО ВЫ ХОТИТЕ ИЗМЕНИТЬ?');READLN(IM);
IF IM='A ' THEN BEGIN WRITE('A=<');/*ВС
        MOV #^O7714,AME
        MOVB DME,A(5)
*/ IF SYST THEN WRITE(A:-3,'> ') ELSE BEGIN DATA:=A;OUT16;
WRITE('> ');END;I816;A:=ADR;/*ВС
        MOV A(5),DME
*/;END ELSE
IF IM='B ' THEN BEGIN WRITE('B=<');/*ВС
        MOV #^O7712,AME
        MOVB DME,B(5)
*/ IF SYST THEN WRITE(B:-3,'> ') ELSE BEGIN DATA:=B;OUT16;
WRITE('> ');END;I816;B:=ADR;/*ВС
        MOV B(5),DME
*/;END ELSE
IF IM='C ' THEN BEGIN WRITE('C=<');/*ВС
        MOV #^O7711,AME
        MOVB DME,C(5)
*/ IF SYST THEN WRITE(C:-3,'> ') ELSE BEGIN DATA:=C;OUT16;
WRITE('> ');END;I816;C:=ADR;/*ВС
        MOV C(5),DME
*/;END ELSE
IF IM='D ' THEN BEGIN WRITE('D=<');/*ВС
        MOV #^O7710,AME
        MOVB DME,D(5)
*/ IF SYST THEN WRITE(D:-3,'  ') ELSE BEGIN DATA:=D;OUT16;
WRITE('> ');END;I816;D:=ADR;/*ВС
        MOV D(5),DME
*/;END ELSE
IF IM='E ' THEN BEGIN WRITE('E=<');/*ВС
        MOV #^O7707,AME
        MOVB DME,E(5)
*/ IF SYST THEN WRITE(E:-3,'> ') ELSE BEGIN DATA:=E;OUT16;
WRITE('> ');END;I816;E:=ADR;/*ВС
        MOV E(5),DME
*/;END ELSE
IF IM='H ' THEN BEGIN WRITE('H=<');/*ВС
        MOV #^O7706,AME
        MOVB DME,H(5)
*/ IF SYST THEN WRITE(H:-3,'> ') ELSE BEGIN DATA:=H;OUT16;
WRITE('> ');END;I816;H:=ADR;/*ВС
```

```
        MOV H(5),DME
*/:END ELSE
IF IM='L ' THEN BEGIN WRITE('L=');/*BC
        MOV #^O7705,AME
        MOVB DME,L(5)
*/ IF SYST THEN WRITE(L:-3,' ') ELSE BEGIN DATA:=L;OUT16;
WRITE('  ');END;I816;L:=ADR;/*BC
        MOV L(5),DME
*/:END ELSE
IF IM='SP' THEN BEGIN WRITE('SP=');/*BC
        MOV #^O7715,AME
        MOVB DME,SP(5)
        MOV #^O7716,AME
        MOVB DME,SP+1(5)
*/IF SYST THEN WRITE(SP:-6,' ') ELSE BEGIN DATA:=SP;OUT16;
WRITE('  ');END;I816;SP:=ADR;/*BC
        MOVB SP+1(5),DME
        DEC AME
        MOVB SP(5),DME
*/:END ELSE BEGIN
WRITELN('ТАКОГО РОН В ЦП КР580ВМ80 НЕТ!!!');
END;END;
/*============================*/
PROCEDURE RON;
LABEL 1,2;
VAR I:INT;
BEGIN RS:=5;US:=RS AND 177000B;O:=AME:AME:=0;
RS:=US OR 6400B;
I:=RS AND 50B;IF I=40B THEN
 BEGIN SS:=RS AND 57400B;RS:=SS;SS:=SS OR 16000B;RS:=SS;JUMP;
 STORE;SPECPR;AME:=O;RS:=SS AND 177000B;FOR I:=1 TO 10 DO RS:=6;
1: I:=RS AND HUND;IF I=0 THEN GOTO 1;
 O:=AME;AME:=O;RS:=SS OR 400B;MODREG;AME:=O;RS:=SS AND 177000B;
 FOR I:=1 TO 7 DO RS:=6; 2: I:=RS AND HUND;
 IF I=0 THEN GOTO 2;O:=AME;AME:=0;RS:=SS OR 400B;RESTORE;
 END ELSE BEGIN
 WRITELN('ИЗМЕНИТЬ СОДЕРЖИМОЕ РОН НЕВОЗМОЖНО!');
 AME:=0;RS:=US;GOTO 999;END;AME:=0;RS:=US;
END;
/*============================*/
PROCEDURE CONF;
LABEL 2,3,4,5;
 PROCEDURE ILANS;
 BEGIN WRITELN('ОТВЕТ НЕВЕРЕН,ПОВТОРИТЕ!');
 END;
BEGIN US:=RS AND 177400B;
WRITELN('ОПРЕДЕЛИТЕ КОНФИГУРАЦИЮ СИСТЕМЫ.');
2: WRITE('0-ПАМЯТЬ ПОЛЬЗОВАТЕЛЯ,1-ПАМЯТЬ ЭМУЛЯТОРА-?');
READLN(ANS);IF ANS=0 THEN BEGIN US:=US AND 135400B;
US:=US OR MEMORY;GOTO 4;END ELSE
IF ANS=1 THEN US;=US AND 77400B ELSE
BEGIN ILANS;GOTO 2;END;
3: WRITE('0-ВНЕШНИЙ ГЕНЕРАТОР,1-ГЕНЕРАТОР ЭМУЛЯТОРА-?');
READLN(ANS);IF ANS=0 THEN BEGIN US:=US AND 135400B; GOTO 4;END
ELSE IF ANS=1 THEN US:=US OR GENER ELSE
BEGIN ILANS;GOTO 3;END;
5: WRITE('ВХОДИТ ЛИ МПУ В КОНФИГУРАЦИЮ СИСТЕМЫ(0-НЕТ,1-ДА)?');
READLN(ANS);
IF ANS=0 THEN US:=US OR 2000B ELSE
IF ANS=1 THEN US:=US AND 175400B ELSE
BEGIN ILANS;GOTO 5;END;
4: RS:=US;END;
/*============================*/
PROCEDURE RUN;
```

```
VAR TA,LEN:INT;
BEGIN  LEN:=1;REPEAT
WRITE('ВВЕДИТЕ ИМЯ ПРОГРАММЫ ПОЛЬЗОВАТЕЛЯ-');
READLN(NAME);RESET(FTXT,NAME,'OBJ',LEN);
UNTIL LEN<>-1;
RD:AZAG:=(RAB MOD 256)*256;
RD:AZAG:=AZAG+(RAB MOD 256);
WRITE('АДРЕС ЗАГРУЗКИ=');IF SYST THEN WRITELN(AZAG:-6) ELSE
BEGIN DATA:=AZAG;OUT16;WRITELN;END;
RD:DMOD:=(RAB MOD 256)*256;
RD:DMOD:=DMOD+(RAB MOD 256);
WRITE(' ДЛИНА МОДУЛЯ=');IF SYST THEN WRITELN(DMOD:-6) ELSE
BEGIN DATA:=DMOD;OUT16;WRITELN;END;TA:=0;
WHILE TA<>DMOD DO
BEGIN RD;K:=RAB MOD 256;NEPREG(K,AZAG);
AZAG:=AZAG+1;TA:=TA+1;
END;END;
/*==========================*/
PROCEDURE USTKT;
LABEL 1;
BEGIN US:=RS AND 177000B;RS:=5;D:=AME;AME:=0;RS:=US OR 6400B;
WRITELN('ВВЕДИТЕ АДРЕСА,В КОТОРЫЕ ВЫ ХОТИТЕ УСТАНОВИТЬ КОНТР.ТОЧКИ.');
WRITELN('ПРИЗНАК ОКОНЧАНИЯ ВВОДА КОНТР.ТОЧЕК-ОВ ИЛИ ОН');
1: WRITE('АДРЕС КОНТР.ТОЧКИ?');I816;IF ADR <>0 THEN BEGIN /*BC
        MOV ADR(5),AME
        MOV DME,DATA(5)
        BIS #^O400,DATA(5)
        MOV DATA(5),DME
*/;GOTO 1;END ELSE AME:=D;RS:=US;
END;
/*==========================*/
PROCEDURE NOKT;
BEGIN US:=RS AND 177000B;RS:=5;D:=AME;AME:=0;RS:=US OR 6400B;
/*BC
        MOV MAX(5),R0
        CLR R1
NOK0:   MOV R1,AME
        MOV DME,R2
        BIC #^O177400,R2
        MOV R2,DME
        INC R1
        SOB R0,NOK0
        MOV #^O10000,R0
        MOV SDV(5),R1
NOK1:   MOV R1,AME
        MOV DME,R2
        BIC #^O177400,R2
        MOV R2,DME
        INC R1
        SOB R0,NOK1
*/;
AME:=0;RS:=US;END;
/*==========================*/
PROCEDURE REGEMUL;
LABEL 22;
BEGIN US:=RS AND 177000B;D:=AME;AME:=0;RS:=US OR 400B;
SS:=RS AND 177400B;
22: WRITELN('НАЗНАЧЕНИЕ РЕЖИМА ЭМУЛЯЦИИ');
WRITELN('0-НЕПРЕРЫВНЫЙ');
WRITE('1-ПОШАГОВЫЙ-');
READLN(ANS);CASE ANS OF
0: BEGIN RS:=SS AND 177400B;SHAG:=FALSE;END;
1: BEGIN RS:=SS OR 5R;SHAG:=TRUE;END;
ELSE GOTO 22
```

```
END;RUN;US:=RS AND 177000B;
RS:=US;END;
/*=========================*/
PROCEDURE PUNKT1;
BEGIN
 IF (DME AND MEMORY)=0 THEN
 WRITELN('ИДЕТ ВЫПОЛНЕНИЕ ПРОГРАММЫ.') ELSE
 WRITELN('ЦП В ЗАХВАТЕ.');GOTO 999;
END;
/*=========================*/
PROCEDURE PRODEMUL;
BEGIN PC0:=PC;RS:=6;IF SHAG>REG THEN MNEM;
END;
/*=========================*/
PROCEDURE PUSKEMUL;
LABEL 22;
VAR C0,C1:INT;
BEGIN
WRITE('ВВЕДИТЕ АДРЕС ПУСКА ПРОГРАММЫ-');I816;
22: WRITE('РАЗРЕШЕНА РАБОТА С ПРЕРЫВАНИЯМИ(0-НЕТ,1-ДА)?');READLN(ANS);
IF ANS=0 THEN COM1:=363B ELSE
IF ANS=1 THEN COM1:=373B ELSE
BEGIN WRITELN('ОТВЕТ НЕВЕРЕН,ПОВТОРИТЕ');GOTO 22;END;
IF (DME AND HUND)=0 THEN PUNKT1 ELSE
BEGIN US:=RS AND 177000B;RS:=5;
Q:=AME;AME:=Q;SS:=US OR 2000B;RS:=SS;SS:=SS AND 57400B;
RS:=SS;SS:=SS OR 16000B;RS:=SS;SS:=SS OR 400B;RS:=SS;
PC0:=ADR;/*ВС
        MOV #0,AME
        MOV DME,C0(6)
        MOV COM1(5),DME
        INC AME
        MOV DME,C1(6)
        MOV #^O303,DME
        INC AME
        MOV DME,COM2(5)
        MOVB ADR(5),DME
        INC AME
        MOV DME,COM3(5)
        MOVB ADR+1(5),DME
        MOV Q(5),AME
        BIC #^O400,SS(5)
        MOV SS(5),RS
        MOV #4,RS
        BIT #^O1000,DME
        BEQ .-6
        MOV #6,RS
        BIT #^O1000,DME
        BEQ .-6
        MOV #6,RS
        BIT #^O1000,DME
        BEQ .-6
        MOV AME,Q(5)
        MOV Q(5),AME
        BIS #^O400,SS(5)
        MOV SS(5),RS
        MOV #3,AME
        MOV COM3(5),DME
        DEC AME
        MOV COM2(5),DME
        DEC AME
        MOV C1(6),DME
        DEC AME
        MOV C0(6),DME
```

```
*/;AME:=0;RS:=US;RS:=/;
IF SHAG REG THEN MNEM;ENU;ENU;
/*=========================== */
PROCEDURE SYSTIS;
LABEL 1;
BEGIN WRITELN('В КАКОЙ СИСТЕМЕ СЧИСЛЕНИЯ ВЫ ХОТИТЕ РАБОТАТЬ');
1: WRITELN('0 - ВОСЬМЕРИЧНАЯ СИСТЕМА');
WRITE('1 - ШЕСТНАДЦАТЕРИЧНАЯ СИСТЕМА-');READLN(ANS);
IF ANS=0 THEN BEGIN SYST:=TRUE;WRITE('ВЫБРАНА ВОСЬМЕРИЧНАЯ СИСТЕМА');
WRITELN(' СЧИСЛЕНИЯ');END ELSE
IF ANS=1 THEN BEGIN SYST:=FALSE;Z16:=FALSE;
WRITELN('ВЫБРАНА ШЕСТНАДЦАТЕРИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ');
END ELSE BEGIN WRITELN('ОТВЕТ НЕВЕРЕН, ПОВТОРИТЕ');GOTO 1;
END;END;
/*=========================== */
PROCEDURE SHOW;
LABEL 11;
BEGIN
WRITE('ПАМЯТЬ ');
IF (RS AND MEMORY)=0 THEN WRITE('ЭМУЛЯТОРА') ELSE
WRITE('ПОЛЬЗОВАТЕЛЯ');WRITE('  :5);
WRITE('ГЕНЕРАТОР ');
IF (RS AND GENER)=0 THEN WRITE('МПУ') ELSE WRITE('ЭМУЛЯТОРА');
WRITE(' ':5);
IF (RS AND CR)=0 THEN 0:=0 ELSE
 BEGIN WRITELN('ПОЦИКЛОВЫЙ РЕЖИМ');GOTO 11;END;
IF (RS AND SR)=0 THEN WRITELN('НЕПРЕРЫВНЫЙ РЕЖИМ') ELSE
WRITELN('ПОШАГОВЫЙ РЕЖИМ');
11: OPROS;IF(DME AND 2000B)=0 THEN
 BEGIN WRITELN('ОСТАНОВ ПО НЕГОТОВНОСТИ МПУ.');GOTO 999;END;
END;
/*=========================== */
PROCEDURE TRACE;
VAR J,K:INT;
    SH:BOOLEAN;
BEGIN SH:=SHAG;SHAG:=FALSE;PUSKEMUL;
        FOR K:=1 TO 13 DO
        BEGIN FOR J:=1 TO 10 DO
        BEGIN OPROS;PC:=AME;
 IF SYST THEN WRITE(PC:-6,' ') ELSE BEGIN DATA:=PC;
 OUT16;WRITE('  ');END;RS:=6;
        END;
        WRITELN;END;SHAG:=SH;
END;
/*=========================================== */
PROCEDURE LOADRAM;
LABEL 1;
VAR AN,AK,COM,CO:INT;
    ER,DE,DU,I:INT;
BEGIN ER:=0;
WRITE('НАЧАЛЬНЫЙ АДРЕС ЗАГРУЖАЕМОГО МАССИВА ДАННЫХ?');I816;AN:=ADR;
WRITE('КОНЕЧНЫЙ АДРЕС ЗАГРУЖАЕМОГО МАССИВА ДАННЫХ?');I816;AK:=ADR+1;
COM:=AK-AN;US:=RS AND 177000B;RS:=5;0:=AME;AME:=0;SS:=US AND 57400B;
SS:=SS OR CR;RS:=SS;RS:=SS OR 400B;PC:=AME;
COM1:=DME;DME:=303B;AME:=AME+1;COM2:=DME;DME:=320B;
AME:=AME+1;COM3:=DME;DME:=17B;AME:=0;RS:=SS AND 177000B;
RS:=6; RS:=6; RS:=6;
1: I:=RS AND HUND;IF I=0 THEN GOTO 1;0:=AME;AME:=0;RS:=SS OR 400B;
AME:=PC;DME:=COM1;AME:=AME+1;DME:=COM2;AME:=AME+1;DME:=COM3;
STORE;/*ВС
        MOV #^O7720,R0
        MOV #15,R1
        MOV #LD0,R2
LD1:    MOV R0,AME
```

```
        MOV (R2)+,DME
        INC R0
        SOB R1,LD1
        BR LD2
LD0:    .WORD ^042,^0305,^017,^062,^0314,^017, 052,^0316
        .WORD ^017,^0176,^0167,^043,^0303,^0331,^017
LD2:    MOV #^07716,AME
        MOVB AN(6),DME
        INC AME
        MOVB AN+1(6),DME
        MOV 0(5),AME
        MOV SS(5),RS
        MOV #17,R0
LD3:    MOV #6,RS
        SOB R0,LD3
        MOV SS(5),DE(6)
        BIC #^052000,DE(6)
        BIS #^0100000,DE(6)
        MOV DE(6),RS
        MOV #6,RS
        DEC COM(6)
        BEQ LD4
        MOV SS(5),RS
        MOV #7,R0
        BR LD3
LD4:    MOV SS(5),RS
        MOV AME,C0(6)
        MOV C0(6),AME
        BIS #^0400,SS(5)
        MOV SS(5),RS
        MOV #^07733,R0
        MOV #7,R1
        MOV #LD10,R2
LD8:    MOV R0,AME
        MOV (R2)+,DME
        INC R0
        SOB R1,LD8
        BR LD9
LD10:   .WORD ^072,^0314,^017,^052,^0305,^017,^0303
LD9:    MOV R0,AME
        MOVB PC(5),DME
        INC AME
        MOVB PC+1(5),DME
        MOV AME,C0(6)
        MOV C0(6),AME
        BIC #^0400,SS(5)
        MOV SS(5),RS
        MOV #12,R0
LD11:   MOV #6,RS
        SOB R0,LD11
        MOV AME,C0(6)
        MOV C0(6),AME
        BIS #^0400,SS(5)
        MOV SS(5),RS
        SUB AN(6),AK(6)
        MOV AK(6),R0
LD5:    MOV AN(6),AME
        MOV DME,DE(6)
        MOV DMU,DU(6)
        BIC #^0177400,DE(6)
        BIC #^0177400,DU(6)
        CMP DE(6),DU(6)
        BNE LD6
        INC AN(6)
```

```
        SOB R0,LD5
        BR LD7
LD6:    MOV #1,ER(6)
LD7:    NOP
*/;RESTORE;IF ER=1 THEN
BEGIN WRITE('ПРИ ЗАГРУЗКЕ ПАМЯТИ МПУ ПРОИЗОШЕЛ СБОЙ.');
WRITELN('ПОВТОРИТЕ ЗАГРУЗКУ.');
END ELSE WRITELN('ЗАГРУЗКА ОЗУ МПУ ЗАВЕРШЕНА.');
AME:=0;RS:=US;END;
/*=====================================*/
PROCEDURE ZAGTT;
LABEL 1,2;
VAR AZ:INT;
BEGIN IF (RS AND SR)=0 THEN SHAG:=FALSE ELSE SHAG:=TRUE;
WRITELN('ВВЕДИТЕ КОДЫ КОМАНД ЧЕРЕЗ  ВК'.КОНЕЦ ВВОДА-777В ИЛИ 1FF');
WRITE('ЗАДАЙТЕ НАЧАЛЬНЫЙ АДРЕС-');
I816;AZ:=ADR;
US:=RS AND 177000B;0:=AME;AME:=0;RS:=US OR 6400B;
1: IF SYST THEN
WRITE(AZ:-6,'/') ELSE BEGIN DATA:=AZ:OUT16;WRITE('/');END;
I816:IF ADR=777B THEN GOTO 2;/*BC
        MOV AZ(6),AME
        MOV ADR(5),DME
*/;AZ:=AZ+1;GOTO 1;
2: RS:=US;END;
/*=====================================*/
PROCEDURE USTVVOD;
LABEL 1;
VAR UST:ARRAY [1..3] OF CHAR;
BEGIN
1: WRITE('С КАКОГО УСТРОЙСТВА БУДЕТ ЗАГРУЖЕНА ПРОГРАММА ?');
READLN(UST);IF UST='TT:' THEN ZAGTT ELSE
IF UST='DX:' THEN REGEMUL ELSE
BEGIN WRITELN('ТАКОГО УСТРОЙСТВА НЕТ ');GOTO 1;END;END;
/*=====================================*/
PROCEDURE HALTSS;
LABEL 1,2,3;
VAR I,J,SUS:INT;
  SH:BOOLEAN;
BEGIN   SUS:=RS AND 177000B;SH:=SHAG;SHAG:=FALSE;
1:      WRITELN('ВВЕДИТЕ КОД СЛОВА СОСТОЯНИЯ');
        врителн('1 - ВЫБОРКА КОДА ОПЕРАЦИИ');
        WRITELN('2 - ЧТЕНИЕ ПАМЯТИ');
        WRITELN('3 - ЗАПИСЬ В ПАМЯТЬ');
        WRITELN('4 - ЧТЕНИЕ СТЕКА');
        WRITELN('5 - ЗАПИСЬ В СТЕК');
        WRITELN('6 - ВВОД ОТ ВНЕШНЕГО УСТРОЙСТВА');
        WRITELN('7 - ВЫВОД НА ВНЕШНЕЕ УСТРОЙСТВО');
        WRITELN('8 - ОБРАБОТКА ПРЕРЫВАНИЯ');
        WRITELN('9 - ПОДТВЕРЖДЕНИЕ ОСТАНОВА');
        WRITE('10 - ПОДТВЕРЖДЕНИЕ ПРЕРЫВАНИЯ ПРИ ОСТАНОВЕ-');
READLN(I);CASE I OF
           1: I:=242B;
           2: I:=202B;
           3: I:=0;
           4: I:=206B;
           5: I:=4;
           6: I:=102B;
           7: I:=20B;
           8: I:=43B;
           9: I:=212B;
          10: I:=53B;
        ELSE GOTO 1 END;
        RS:=SUS OR CR; PUSKEMUL; OPROS; J:=0;SHAG:=SH;
```

```
2: 0:=AME;AME:=0;SS:=RS AND 177000B;RS:=SS OR 400B;
        COM3:=RS AND 377B;
        IF T  COM3 THEN BEGIN 0:=AME;RS:=SS;RS:=A; J:=J+1;END ELSE
                        BEGIN RS:=SUS AND 177000B;GOTO 3;END;
        IF J<=77776B THEN GOTO 2 ELSE
BEGIN RS:=SUS;
WRITELN('УСЛОВИЕ ОСТАНОВА НЕ НАЙДЕНО');
GOTO 999;END;
3: END;
/*====================*/
PROCEDURE NONSTOP;
BEGIN US:=RS AND 177000B;RS:=US AND 157000B;SHAG:=FALSE;
RS:=US AND 147000B;REG:=FALSE;NOKT;
END;
/*====================*/
BEGIN
WRITE(CHR(33B),CHR(105B),CHR(14B),CHR(27B));
WRITELN('  ':25,'-СХЕМНЫЙ ЭМУЛЯТОР КР580ВМ80-');
WRITELN;US:=66000B;RS:=US;REG:=FALSE;SYST:=FALSE;RS:=160001B;
RES;RS:=1;0:=AME;AME:=0;RS:=US OR 400B;
SYSTIS;ZADVOL;CONFMEM;
OCHMEM;US:=RS AND 177400B;RS:=US AND 157400B;
CONF;USTVVOD;0:=AME;AME:=0;US:=RS AND 177400B;RS:=US AND 177000B;
999: WRITE('*');READLN(COM);
IF COM='GO  ' THEN PUSKEMUL ELSE
IF COM='P   ' THEN PRODEMUL ELSE
IF COM='RES ' THEN RES ELSE
IF COM='TYPE' THEN TYPEMEM ELSE
IF COM='MEM ' THEN IZMMEM ELSE
IF COM='CONF' THEN CONF ELSE
IF COM='R   ' THEN RON ELSE
IF COM='CYKL' THEN CIREG ELSE
IF COM='STOP' THEN BEGIN RS:=US OR 20000B;RS:=US AND 157400B;END ELSE
IF COM='SHOW' THEN SHOW ELSE
IF COM='STEP' THEN BEGIN RS:=US OR SR;SHAG:=TRUE;END ELSE
IF COM='KT  ' THEN USTKT ELSE
IF COM='NONS' THEN NONSTOP ELSE
IF COM='TRAC' THEN TRACE ELSE
IF COM='LOAD' THEN LOADRAM ELSE
IF COM='HALT' THEN HALTSS ELSE
IF COM='BOOT' THEN USTVVOD ELSE
IF COM='CR  ' THEN BEGIN SS:=RS AND 177400B;RS:=SS OR CR;REG:=TRUE;
                    END ELSE
BEGIN WRITELN(CHR(7B),'КОМАНДЫ ',COM:4,' НЕ СУЩЕСТВУЕТ');
GOTO 999;END;OPROS;
IF WAIT=TRUE THEN INDINF ELSE GOTO 999;
GOTO 999;
END.
```

ЗАКЛЮЧЕНИЕ

Современные ОС МПУ (отечественные и зарубежные) поддерживают лишь часть этапов проектирования МПУ – проектирование ПС, частично АС (отладка и тестирование) и комплексную отладку и тестирование МПУ. Этапы структурного и логического моделирования МПУ, разработки конструкции, разработки и изготовления фотошаблонов печатных плат, разработки конструкторской и программной документации поддерживаются системами автоматизации проектирования других типов, построенными на базе высокопроизводительных мини- и микроЭВМ, включающих специальные типы устройств ввода-вывода (графические дисплеи с высокой разрешающей способностью, графопостроители, кодировщики и т. п.).

Необходимость в наличии нескольких типов САПР или автоматизированных рабочих мест, используемых в процессе проектирования МПУ и построенных на разнородных технических и программных средствах, приводит к увеличению стоимости разработки, неизбежным ошибкам и нестыковкам при переходе от одного типа САПР к другому, снижению качества и увеличению сроков разработки МПУ.

Из сказанного становится очевидно, что перспективы развития отечественных и зарубежных ОС МПУ связаны с постепенным превращением их в единый тип САПР – САПР МПУ, поддерживающий все этапы проектирования МПУ от разработки технического задания до разработки конструкторской и программной документации и испытаний готового изделия. Для решения этой задачи необходимо решение ряда сложных научных и технических проблем, в том числе:

разработка и организация серийного производства массовых высокопроизводительных персональных рабочих станций, оснащенных необходимыми типами специальных устройств ввода-вывода. Данные рабочие станции должны допускать возможность коллективного использования их ресурсов различными разработчиками МПУ (системотехниками, программистами, конструкторами, схемотехниками, технологами и т. п.), имеющими доступ к единой базе данных;

разработка необходимых системных и прикладных ПС САПР МПУ, ориентированных на поддержку всех этапов проектирования МПУ.

Ближайшие перспективы развития ОС МПУ, в том числе описанных в данной книге универсальных отладочных комплексов, связаны с реализацией следующих основных функциональных возможностей:

расширение состава используемых кросс-трансляторов языков высокого уровня;

отладка многопроцессорных МПУ;

создание на базе УОК сетей проектирования (локальных или распределенных).

СПИСОК ЛИТЕРАТУРЫ

1. **Фаулджер Р.** Программирование встроенных микропроцессоров. М.: Мир, 1985.

2. **Микропроцессоры:** системы программирования и отладки/ В. А. Мясников, М. Б. Игнатьев, А. А. Кочкин и др. М.: Энергоатомиздат, 1985.

3. **Васильев Н. П., Горовой В. Р.** Аппаратурно-программные средства отладки. М.: Высшая школа, 1984.

4. **Мун Д.** Микрокомпьютер для эмуляции// Электроника. 1980. № 16. С. 48—53.

5. **Кобылинский А. В., Липовецкий Г. П.** Однокристальные микроЭВМ серии К1816// Микропроцессорные средства и системы. 1986. № 1. С. 10—19.

6. **Герсон С.** Микрокомпьютер со сменным стираемым ППЗУ — эффективное средство эмуляции однокристальных микрокомпьютеров// Электроника 1980. № 3. С. 34—40.

7. **Однокристальные** микропроцессоры комплекта БИС серии К1801/ В. Л. Джхунян, Ю. И. Борщенко, В. Р. Науменков и др.// Микропроцессорные средства и системы. 1984. № 4. С. 12—18.

8. **Белов А. М., Иванов Е. А., Муренко Л. Л.** Комплексы кросс-программ «Электроника МИКРОСС»// Микропроцессорные средства и системы. 1986. № 3. С. 27.

9. **Попов А. А., Хохлов М. М., Глухман В. Л.** Диалоговые вычислительные комплексы «Электроника НЦ—80-20»// Микропроцессорные средства и системы. 1984. № 4. С. 61—64.

10. **Микропроцессорные** комплекты интегральных схем/ В. С Борисов, А. А. Васенков, Б. М. Малашевич и др. М.: Радио и связь, 1982.

11. **Кобылинский А. В., Москалевский А. И., Темченко В. А.** Однокристальный высокопроизводительный 16-разрядный микропроцессор КМ1810ВМ86// Микропроцессорные средства и системы. 1986. № 1. С. 28—33.

12. **Хоровиц П., Хилл У.** Искусство схемотехники: Пер. с англ. М.: Мир, 1984. Т. 2.

13. **Иванников А. Д.** Аппаратно-программные комплексы для отладки микропроцессорных устройств// Микроэлектроника и полупроводниковые приборы/ Под ред. А. А. Васенкова и Я. А. Федотова. М.: Радио и связь. 1981. Вып. 6. С. 62—82.

14. **Универсальная** система программирования БИС/ В. В. Андреев, С. А. Котрелев, А. Н. Филимонов, В. М. Чурков// Электронная промышленность. 1986. Вып. 9. С. 50.

55 к.

В СЕРИЮ

„МИКРОПРОЦЕССОРНЫЕ БИС И ИХ ПРИМЕНЕНИЕ"

ВКЛЮЧЕНЫ КНИГИ:

1. *Басманов А. С., Широков Ю. Ф.* Микропроцессоры и однокристальные микро ЭВМ: Номенклатура и функциональные возможности.
2. *Домрачев В. Г., Иванов С. Н., Романов А. Ф., Чернышов Ю. Н.* Одноплатные микро ЭВМ.
3. *Гамкрелидзе С. А., Завьялов А. В., Мальцев П. П., Соколов В. Г.* Цифровая обработка информации на основе быстродействующих БИС.
4. *Муренко Л. П., Чурков В. М., Широков Ю. Ф.* Программаторы запоминающих и логических интегральных микросхем.
5. *Домнин С. Б., Иванов Е. А., Муренко Л. П.* Средства комплексной отладки микропроцессорных устройств.
6. *Белов А. М., Иванов Е. А., Муренко Л. П.* Средства автоматизации программирования микропроцессорных устройств.
7. *Овчинников В. В.* Архитектура распределенных информационно-вычислительных микропроцессорных систем.
8. *Бойченко Е. В.* Методы схемотехнического проектирования распределенных информационно-вычислительных микропроцессорных систем.

55 к.

В СЕРИЮ

„МИКРОПРОЦЕССОРНЫЕ БИС И ИХ ПРИМЕНЕНИЕ"

ВКЛЮЧЕНЫ КНИГИ:

1. *Басманов А. С., Широков Ю. Ф.* Микропроцессоры и однокристальные микро ЭВМ: Номенклатура и функциональные возможности.
2. *Домрачев В. Г., Иванов С. Н., Романов А. Ф., Чернышов Ю. Н.* Одноплатные микро ЭВМ.
3. *Гамкрелидзе С. А., Завьялов А. В., Мальцев П. П., Соколов В. Г.* Цифровая обработка информации на основе быстродействующих БИС.
4. *Муренко Л. Л., Чурков В. М., Широков Ю. Ф.* Программаторы запоминающих и логических интегральных микросхем.
5. *Домнин С. Б., Иванов Е. А., Муренко Л. Л.* Средства комплексной отладки микропроцессорных устройств.
6. *Белов А. М., Иванов Е. А., Муренко Л. Л.* Средства автоматизации программирования микропроцессорных устройств.
7. *Овчинников В. В.* Архитектура распределенных информационно-вычислительных микропроцессорных систем.
8. *Бойченко Е. В.* Методы схемотехнического проектирования распределенных информационно-вычислительных микропроцессорных систем.